Computers in Biology and Medicine 127 (2020) 104096

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: http://www.elsevier.com/locate/compbiomed

Hi-GCN: A hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction

Hao Jiang [a], Peng Cao [a,b,*], MingYi Xu [a], Jinzhu Yang [a], Osmar Zaiane [c]

[a] Computer Science and Engineering, Northeastern University, Shenyang, China
[b] Key Laboratory of Intelligent Computing in Medical Image of Ministry of Education, Northeastern University, Shenyang, China
[c] Amii, University of Alberta, Edmonton, Alberta, Canada

ARTICLE INFO

Keywords:
Alzheimer's disease
Autism spectrum disorder
Brain network
Graph classification
Graph convolutional network

ABSTRACT

Purpose: Recently, brain connectivity networks have been used for the classification of neurological disorder, such as Autism Spectrum Disorders (ASD) or Alzheimer's disease (AD). Network analysis provides a new way for exploring the association between brain functional deficits and the underlying structural disruption related to brain disorders. Network embedding learning that aims to automatically learn low-dimensional representations for brain networks has drawn increasing attention in recent years.

Method: In this work we build upon graph neural network in order to learn useful representations for graph classification in an end-to-end fashion. Specifically, we propose a hierarchical GCN framework (called hi-GCN) to learn the graph feature embedding while considering the network topology information and subject's association at the same time.

Results: To demonstrate the effectiveness of our approach, we evaluate the performance of the proposed method on the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset and Autism Brain Imaging Data Exchange (ABIDE) dataset. Extensive experiments on ABIDE and ADNI datasets have demonstrated competitive performance of the hi-GCN model. Specifically, we obtain an average accuracy of 73.1%/78.5% as well as AUC of 82.3%/86.5% on ABIDE/ADNI. The comprehensive experiments demonstrate that our hi-GCN is effective for graph classification with brain disorders diagnosis.

Conclusion: The proposed hi-GCN method performs the graph embedding learning from a hierarchical perspective while considering the structure in individual brain network and the subject's correlation in the global population network, which can capture the most essential embedding features to improve the classification performance of disease diagnosis. Moreover, the proposed jointly optimizing strategy also achieves faster training and easier convergence than both the hi-GCN with pre-training and two-step supervision.

1. Introduction

Brain disorders such as Alzheimer's disease (AD) [1–4] and Autism spectrum disorder (ASD) [5,6] are considered in terms of disruptions of the normal-range operation of brain functions. While psychiatric disorders are diagnosed based on symptom scores from clinical interview, there are no existing gold standards that can be used for definitive validation. Resting state functional MRI (rs-fMRI) provides us with information about the default state of the brain, and allows us to evaluate functional connectivity and its alterations in brain disorders. It is a method used to evaluate regional interactions that occur in a resting state when an explicit task is not being performed. The patients with ASD or AD dementia exhibit alterations of functional cortical connectivity in rs-fMRI analyses. Brain functional connectivity (FC) derived from rs-fMRI data has become a powerful approach to measure and map brain activity. Recent studies have shown that rs-fMRI based analysis of brain connectivity is effective in helping understand the pathology of brain diseases, such as autism spectrum disorder (ASD) and Alzheimer's disease (AD) [1].

The rs-fMRI data has complex structure, which are inherently represented as a network with a set of nodes and edges [7]. Many works focus on modeling the whole brain rs-fMRI as a network [8–10] and extracting representation from network [11–13]. Recently, functional connectivity networks constructed from rs-fMRI hold great promise for

* Corresponding author. Computer Science and Engineering, Northeastern University, Shenyang, China.
E-mail address: caopeng@cse.neu.edu.cn (P. Cao).

https://doi.org/10.1016/j.compbiomed.2020.104096
Received 1 July 2020; Received in revised form 24 October 2020; Accepted 24 October 2020
Available online 3 November 2020
0010-4825/

distinguishing brain disorder patients from NC(Normal control) [14,15]. It can be regarded as a graph classification problem. It is important to extract a useful network representation from the brain network to facilitate a range of learning tasks, such as brain network classification and network visualization [16,17]. The commonly used features are calculated based on graph-theoretic analysis, such as clustering coefficients and local clustering coefficients [14], which are calculated based on each ROI's local connectivity pattern. However, hand-crafted graph features may not be precise enough to represent or characterize the network [18,19]. In recent years, the high-level feature representation of deep convolutional neural networks has proven to be superior to hand-crafted low-level and mid-level features [20–22]. However, convolutional neural networks and recurrent neural networks, have mainly focused on the grid-structured inputs rather than network structure data.

Network embedding [23–25] is an approach that is used to transform nodes in the network into a lower dimensional representation whilst maximally preserving the network structure. Embedding a brain network into a meaningful low-dimensional representation can improve the classification performance of disease diagnosis. Kipf & Welling [26] proposed graph convolutional networks (GCN) as an effective graph embedding model that naturally combines structure information and node features in the learning process. Recent works have applied GCN on the functional network derived from rs-fMRI data to extract latent features from graph [27–30]. However, the previous works on network embedding learning of brain functional network consider each instance (subject) independently in the learning process, ignoring the association among instances (subjects). The association among the instances (subjects) is critical and the neighborhood information should be considered during the embedding learning. Incorporating and preserving the intrinsic association can promote to learn a better embedding of the brain functional network. The classification methods with modeling the correlation among instances provide a natural framework to analyze the relationships among the instances in the data set. In this framework, nodes can represent individuals (patients or healthy controls) accompanied by a set of features, while the graph edges incorporate associations between subjects. The edge between individuals is implicitly obtained by calculating pairwise similarities. Although no explicit association between nodes exists, the correlation between nodes can be implicitly obtained by calculating pairwise similarities to improve the classification performance.

In our work, we formulate disease diagnosis as a graph classification problem and attempt to advance deep learning for graph-structured data with GCN. To consider the correlation among the samples in population, the population is treated as a network and the aim is to learn network embedding for subjects based on the graph structure. Accurate learning of the network embedding with correlation and estimation of the subject's correlation play critical roles of population network analysis in improving the performance of graph data classification. We propose an end-to-end network embedding learning network for brain functional connectivity networks from both perspective of network level and population level. The framework can analyze and classify the functional connectivity in fMRI data directly in an end-to-end trainable fashion. In the present work, the aim is to achieve a mapping function from a brain network to a low-dimensional vector representation by preserving both topology structure within population network and the structure within individual brain functional network.

To learn the network feature embedding considering the network topology information and subject's association at the same time, two major challenges in building up such a joint graph analysis framework are: how to build an unified framework for network embedding learning and how to optimize the end-to-end network embedding framework? To solve the two challenges, we proposed an effective learning strategy for modeling the brain connectivity network and population network simultaneously in a hierarchical GCN framework (called hi-GCN). The framework involves two individual GCNs for modeling the graph of individual brain functional network and the whole population network, respectively. A compact representation of brain functional network can be learned automatically by a graph-level embedding learning GCN, we called it f-GCN. Then, another GCN, called p-GCN, further updates each node's embedding of the graph data, via aggregating the representations of its neighbors and itself. During the training of p-GCN on the population network, a graph kernel is used directly to measure the similarity between pairs of brain networks. It can capture local properties considering the graph structure when calculating the similarity of nodes with network structure. Through the joint learning of hi-GCN, a high-level embedding of brain network representation can be effectively learned by deriving the structure of brain regions and aggregating embedding of its neighboring subjects in an end-to-end fashion with a global supervision such that the embedding learned is useful for classification.

The main contributions of this paper can be summarized as follows:

1. In this paper, we focus on learning deep representations from fMRI brain connectivity networks, where each brain network represents the brain activity patterns of a particular subject. We propose a Hierarchical GCN (hi-GCN) model which is capable of jointly learning the graph embedding from both the aspects of the brain functional network and the population network at the same time, which can extract the most useful spatial features and capture the most essential embedding features coherently. We assume that the brain network exhibit s network structures at two levels: 1) the region-to-region brain activity correlations in the brain network, and 2) the subject-to-subject relationship in the population network. The hierarchical structural patterns is crucial for learning more accurate representations of the brain network. Specifically, our hi-GCN model has a hierarchical architecture with a two-level GCN. Its first level is to learn a network embedding from the topological structure of the original connectivity network. Its second level, on the other hand, is to incorporate contextual correlation among the subjects to enhance the semantic information. Through the joint learning of hi-GCN, a high-level embedding of brain network representation can be effectively learned by deriving the structure of brain regions and aggregating embedding of its neighboring subjects in an end-to-end fashion with a global supervision such that the embedding learned is useful for classification. With the two-level GCN architecture, each brain network is mapped into an embedding.
2. We develop an optimization strategy for jointly learning the proposed hierarchical GCN model to handle network data with more efficiency. The source code for the proposed architecture is publicly available at: *https://github.com/hao jiang1/hi-GCN*.
3. Extensive experiments on two real medical clinical applications: diagnosis of ASD and diagnosis of Alzheimer's disease, showing the effectiveness of the proposed framework. The experimental results demonstrate that network embedding learning from both correlations within individual brain network and global population network can improve prediction performance.

The rest of the paper is organized as follows. In Section 2, we provide a introduction of fMRI network and GCN. A detailed mathematical formulation of Hierarchical GCN is provided in Section 3. In Section 4, we conducted extensive experiments to verify the advantage of our method for diagnosis of ASD and diagnosis of Alzheimer's disease. The conclusion is drawn in Section 5.

2. Preliminaries

2.1. Problem setup

We define an undirected graph for each subject, $\boldsymbol{N}_i = \{\mathscr{R}_i, \mathscr{A}_i\}$, where $\mathscr{R}_i = \{r_i^1, \ldots, r_i^M\}$ is the set of M nodes, and $\mathscr{A}_i \in R^{M\times M}$ is the

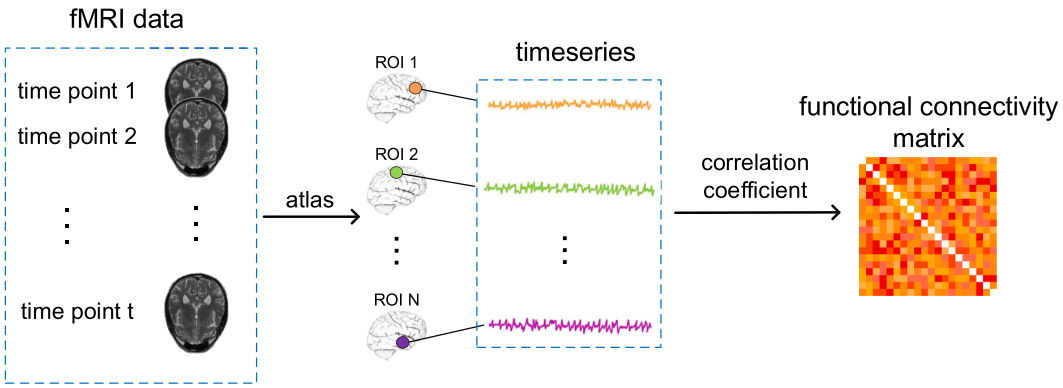


Fig. 1. The procedure of brain FC network construction.

adjacency matrix describing the network's connectivity in the *i*-th subject, where M is the number of ROI. Here $M = 116$. The embedding of each vertex in $\mathscr{R}$ is learned during the GCN training, therefore the initial value of $\mathscr{R}_i$ is set to be one. In the graph classification setting, we have a set of graphs $\{N_1,\ldots,N_D\}$, where D is the size of dataset. Each graph N_i is associated with a label y_i.

In such setting of the population analysis, each subject acquisition is represented by a node and the pairwise similarities are modeled via edges connecting the nodes. Given a collection of images modeled as graphs N_i and the associated label y_i, we construct a global population network $\widehat{N} = \{\widehat{\mathscr{R}}, \widehat{\mathcal{A}}\}$, where $\widehat{\mathcal{A}}$ is the adjacency matrix describing the pairwise similarities between each pair of subjects with brain networks. Each subject is represented by a vertex $\hat{r}$ and is associated with a network data. The definition of the graph's edges is critical in order to capture the underlying structure of the data and explain the similarities between each pair of the N. We employ a graph kernel to estimate the $S(N_i, N_j)$ between two network input of subjects. We will introduce it in Section 3.4. The diagnosis with brain functional network is a typical graph classification problem. The task of the graph classification is to take the brain network as input and predict its corresponding label (clinical status), i.e. patient with disorder or normal control. The aim is to learn the most essential embedding by taking full advantage of the correlation and structure within the graph and accurately predict the label of a given network.

2.2. Functional connectivity network

Recent advances in neuroimaging have led to significant improvements in the spatial resolution of fMRI. As a result, fMRI is currently a widely-used tool that measures brain activity by detecting changes associated with blood flow. The network of the brain can be modeled as a graph consisting of brain regions as the nodes and their functional connectivity as the edges. Brain connectivity networks are widely-used to model inter-regional functional connections, and are typically inferred from the input fMRI data of each subject.

The construction of brain network from fMRI involves two steps, which is shown in Fig. 1. At first, the mean time series for a set of regions extracted from the automated anatomical labeling (AAL) atlas [31] are computed and normalised to zero mean and unit variance. Then, we compute the region-to-region brain correlations by choosing a metric for measuring similarity. Pearson's correlation (PC) analysis between BOLD fMRI signals of any pair of ROIs is the most popular network construction method. Each vertex r_i represents a brain region, and the corresponding time series is indicated as v_i. Pearson correlation coefficient between the fMRI time series v_i at the vertex i and the fMRI time series v_j at the vertex j is given by

$$Q(r_i, r_j) = \frac{Cov(v_i, v_j)}{\sigma_{v_i}\sigma_{v_j}} \tag{1}$$

where $Cov(v_i, v_j)$ is the cross covariance between v_i and v_j, and σ_v denotes the standard deviation of v.

The final constructed network is a graph where the nodes/vertices represent brain regions and the edges are the region-to-region correlations.

2.3. Graph convolutional networks

The power of deep learning models lies in enabling automatic discovery of latent or abstract higher-level information from high dimensional neuroimaging data, which can be an important step to understand complex mental disorders. However, Convolutional Neural Networks (CNNs) do not generalize to irregular graphs since discretized convolutions are only defined for regular domains. Therefore, we use the spectral approach which provides a well-defined localization operator on graphs to define graph convolutions.

Graph Convolutional Networks (GCN) aim to extend the data representation and classification capabilities of convolutional neural networks, which are highly effective for signals defined on regular Euclidean domains, e.g. image and audio signals, to irregular, graph-structured data defined on non-Euclidean domains. The graph convolution is employed directly on graph structured data to extract highly meaningful patterns and features in the space domain. Formally, we are given the adjacency matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$. GCN is stacked by several convolutional layers and a single convolutional layer can be written as:

$$\boldsymbol{E}^{(l+1)} = \mathbf{ReLu}\left(\tilde{\boldsymbol{D}}^{-1/2}\widehat{\mathcal{A}}\tilde{\boldsymbol{D}}^{-1/2}\boldsymbol{E}^{(l)}\boldsymbol{W}^{(l)}\right), \tag{2}$$

where $\widehat{\mathcal{A}} = \mathcal{A} + \boldsymbol{I}_n$, $\tilde{\boldsymbol{D}}_{ii} = \sum_j \widehat{\mathcal{A}}_{ij}$, $\boldsymbol{W}$ is a trainable weight matrix, $\boldsymbol{E}^{(l+1)}$ are the node embeddings (i.e., "messages" omputed after l steps of the GCN, and the node embeddings $\boldsymbol{E}^{(l)}$ generated from the previous message-passing step.

GCN can be considered as a Laplacian smoothing operator for node features over graph structures. The architecture of GCN consists of a series of convolutional layers, each followed by Rectified Linear Unit (ReLU) activation functions to increase non-linearity. The first hidden layer $\boldsymbol{E}^{(0)}$ is set to the input original node features. All layers share the same adjacency matrix. A full GCN run L iterations of Equation (2) to generate the final output node embeddings, $\boldsymbol{E}^{(L)}$.

To localize the filter and reduce the number of parameters, we employ the Chebyshev Polynomials to approximate the convoluational kernels. The computational complexity can be reduced with K-localized convolutions through the polynomial approximation [32]. By recursively applying a stack of graph convolutions with the 1st-order approximation, K-localized convolutions are computed to exploit the information from the K-order neighborhood of central nodes.

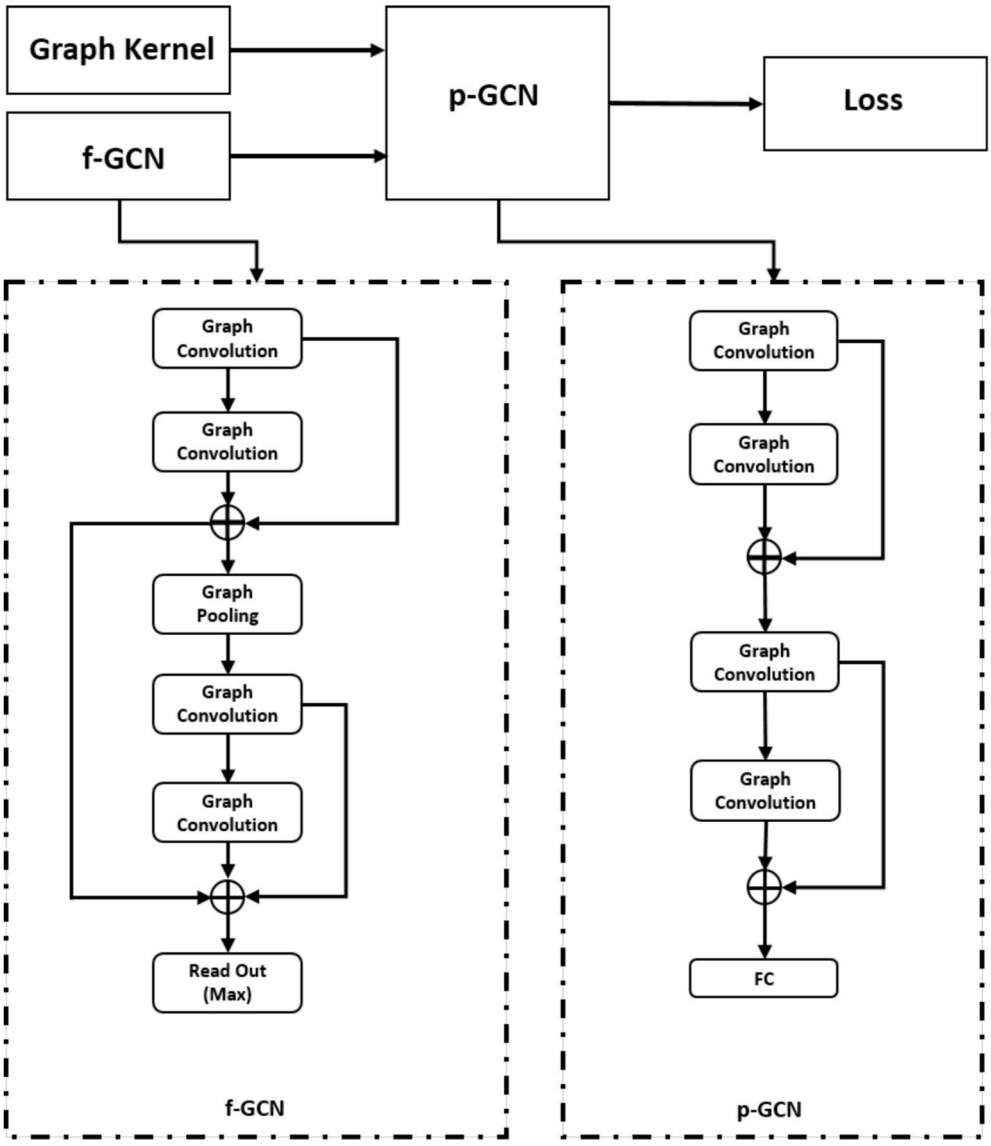


Fig. 2. The architecture of the proposed hi-GCN model for brain network classification.

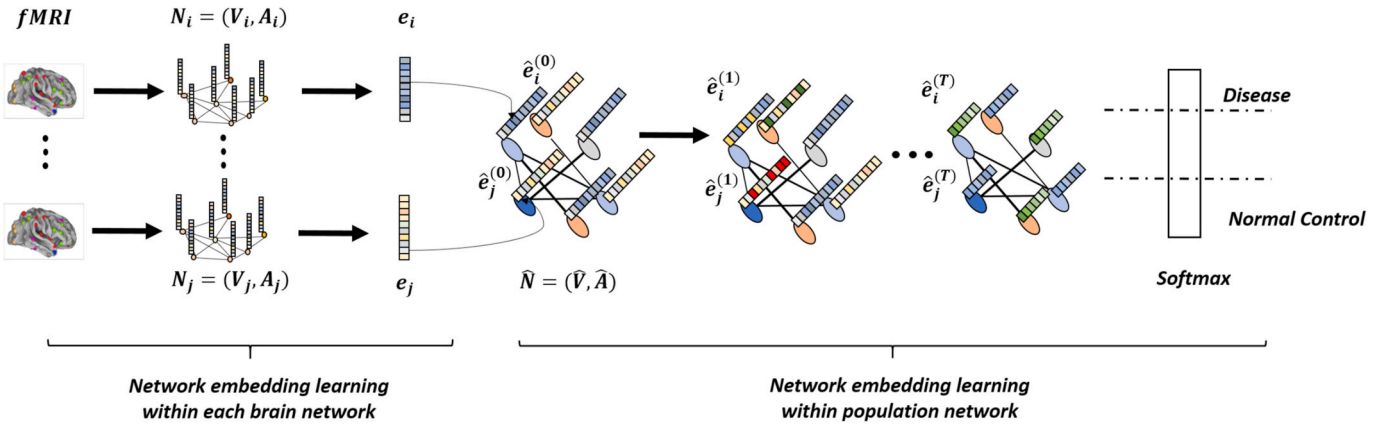


Fig. 3. An illustration of the procedure of network embedding learning in hi-GCN.

3. Hierarchical GCN

3.1. The network architecture of hierarchical GCN

An illustration of the proposed hierarchical graph convolutional networks (hi-GCN) is shown in Fig. 2 for graph representation learning. It consists of the following:

1) f-GCN: learning the latent embedding representation of graph instance based on each ROI's connectivity into a meaningful low-dimensional representation for each brain network instance;
2) Network similarity estimation in the $\widehat{N}$: calculating the network similarity for each pair of network instances with graph kernel;
3) p-GCN: based on the embedding representation learned by f-GCN and the graph pairwise similarity, a new representation of a node is further learned by aggregating the embedding of all neighbors in the population network;
4) Both f-GCN and p-GCN are jointly updated via backpropagation.

The procedure of the embedding learning of the brain functional network is shown in Fig. 3. It includes two phases: network embedding learning within each brain network and embedding learning within the population network. F-GCN produces an embedding $\boldsymbol{E}$ for all network instances, then the learned embedding is fed to the second model (p-GCN) to generate a refined embedding $\widehat{\boldsymbol{E}} \in R^{D\times\widehat{P}}$ where the d-th row describes a latent representation of the brain network from the d-th subject and $\widehat{P}$ is the dimensionality of the final network embedding. Intuitively, $\widehat{\boldsymbol{E}}$ with leveraging the neighborhood data can be used as features for the tasks of brain disorder disease diagnosis. It is important to learn graph representations that can capture rich information from both the fMRI network and the population network. The goal of hi-GCN for the graph classification task is to learn a nonlinear mapping from a brain network to an embedding vector. The procedure is defined as:

$$\mathbf{hi}-\mathbf{GCN}: \boldsymbol{N}\rightarrow\widehat{\boldsymbol{e}}, \tag{3}$$

which involves two functions:

$$\mathbf{f}-\mathbf{GCN}(\boldsymbol{N})=\boldsymbol{e}; \quad \mathbf{p}-\mathbf{GCN}\left(\boldsymbol{e},\widehat{\mathscr{K}}\right)=[\widehat{\boldsymbol{e}},\widehat{y}] \tag{4}$$

By training the entire network end-to-end, the hi-GCN deduces an optimal network embedding for each brain network.

In the next subsection, we introduce details about the two parts of hi-GCN respectively.

3.2. F-GCN

In the f-GCN, multiple GCN layers are stacked. Following the convolutional layers, a global average pooling operator produces coarsened graphs, which can naturally summarize the subgraph information while utilizing the subgraph structure. Moreover, the pooling layers enable GCN to reduce the number of parameters by scaling down the size of the representations, and thus avoid overfitting. A readout layer collapses the node representations of each graph into a graph representation.

When multiple GCN layers are stacked, information about larger neighborhoods are integrated. However, it cannot summarize the node information into the higher level graph representation. To address this challenge, we use [33] to propose eigenvector-based pooling layers EigenPooling to hierarchically summarize node information and generate a graph representation. The pooling layers consist of two components: graph coarsening and generating a coarsened graph by treating subgraphs as supernodes. At first, we adopt spectral clustering to group nodes into subnetworks. The number of clusters is indicated by H, which is further empirically investigated.

Let $\boldsymbol{N}_{coar}$ denote the coarsened graph, which consists of the subnetworks and their connections, $\boldsymbol{C} \in \mathbb{R}^{M\times H}$ ($\boldsymbol{M}$ indicates the number of brain regions) denotes the assignment matrix, which indicates whether a node belongs to a specific subnetwork, and $\mathcal{A}_{coar}$ denote the adjacency matrix of the coarsened subnetworks.

$$\mathcal{A}_{coar}=\sum_{h=1}^{H}\boldsymbol{C}^{(h)}\mathcal{A}_{ext}\left(\boldsymbol{C}^{(h)}\right)^{T}, \tag{5}$$

where $\mathcal{A}_{ext}$ is an adjacency matrix only consisting of the edges between subnetworks.

For the subnetworks, the pooling operator tries to summarize the nodes' features and obtain a representation for the corresponding supernode of the subnetworks. With the structure of the subnetworks, we employ a pooling operator based on the graph spectral theory by facilitating the eigenvectors of the Laplacian matrix of the subnetworks. Let Θ_c denote the pooling operator consisting of all the c-th up-sampled eigenvectors from all the subnetworks.

$$\boldsymbol{E}_c=\Theta_c^{T}\boldsymbol{E} \tag{6}$$

where $\boldsymbol{E}_c \in \mathbb{R}^{H\times P_l}$ (P_l indicates the l-th embedding dimensionality) is the pooled result using the c-th pooling operator.

Finally, we obtain the node features $\boldsymbol{E}_{coar}$ of $\boldsymbol{N}_{coar}$. Instead of using the results pooled by all the pooling operators, we choose to use the first Z of them as $\boldsymbol{E}_{coar}=[\boldsymbol{E}_0,\ldots,\boldsymbol{E}_Z]$ is the final pooled results. With $\mathcal{A}_{coar}$ and $\boldsymbol{E}_{coar}$, we can learn higher level representations of the coarsened graph that exploits the subgraph structure as well as the node features of the input graph.

3.3. p-GCN

p-GCN module further learns the graph embedding by message passing according to 1) the network embedding $\boldsymbol{E}$ describing each subject, and 2) the adjacent matrix between samples $\widehat{\mathcal{A}}$. As shown in Fig. 3, the node features map of the input layer of p-GCN is defined as: $\widehat{\boldsymbol{E}}^{0}=\boldsymbol{E}^{L}$, where $\boldsymbol{E}^{L}$ is the set of node embedding features learned by f-GCN. The main idea is to generate a node $\boldsymbol{N}$'s representation by aggregating its own features $\widehat{\boldsymbol{e}}_i$ and neighbors' features $\widehat{\boldsymbol{e}}_j$, where $j \in \mathbf{Neighbor}(i)$. p-GCN also stack multiple graph convolutional layers to extract high-level node representations. The model inductively learns a node representation by recursively aggregating and transforming feature vectors of its neighboring subjects.

The definition of the graph's edges is critical in order to capture the underlying structure of the data and explain the similarities between the feature vectors. We employ a graph kernel to directly measure the topological similarity between functional connectivity networks. The graph kernel is one kind of kernel constructed on graphs that measures the topological similarity between graphs. More formally, given a pair of networks $\boldsymbol{N}_i$ and $\boldsymbol{N}_j$, a graph kernel can be defined as $S(\boldsymbol{N}_i,\boldsymbol{N}_j)=\langle\varphi(\boldsymbol{N}_i),\varphi(\boldsymbol{N}_j)\rangle$, which takes into account the topology of networks $\boldsymbol{N}_i$ and $\boldsymbol{N}_j$.

Assume we have a function S_I for computing the similarity between the structure of two brain networks rather than the embeddings, and the similarity score between a pair of brain networks $\boldsymbol{N}_i$ and $\boldsymbol{N}_j$ is denoted by $S_I(\boldsymbol{N}_i,\boldsymbol{N}_j)$. Our hi-GCN method employs a graph kernel to estimate the correlation between networks. Kernel methods have the desirable property that they do not rely on explicitly characterizing the vector representation $\varphi(x)$ of data points in the feature space induced by a kernel function, but access data only via the Gram matrix $\mathscr{K}$. In this setting, a kernel $k:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{R}$ is called a graph kernel, which can capture the inherent similarity in the graph structure and is reasonably efficient to evaluate. Distances between instances with the q-th kernel function are calculated and are defined as $\mathscr{K}_q(r_a^i,r_b^i)$, where $r_a^i=\sum_u^M\mathscr{A}_i(a,u)$, indicating the local topology of nodes. We consider each subject as an undirected graph and employ a graph kernel to measure the similarity. We assume that the relation structures of brain network s

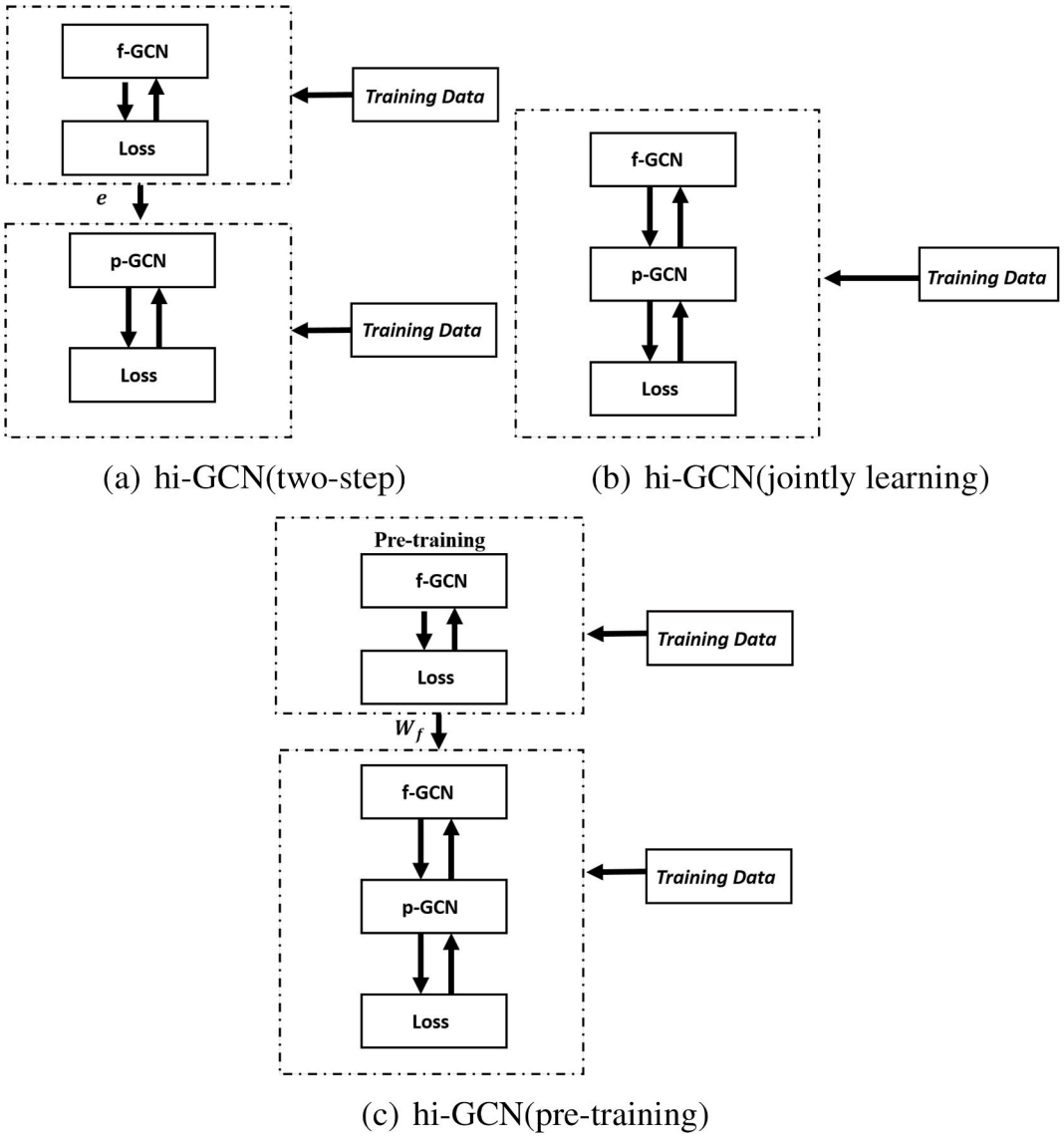


Fig. 4. The different training scheme.

belonging to the same class are relatively more similar, while those belonging to different classes are relatively more dissimilar. Like kernels on vector spaces, graph kernels can be calculated implicitly by computing $\mathscr{K}$. If the RBF kernel function is chosen, then the distance between instances is calculated as: $\mathscr{K}(r_a^i, r_b^i) = \exp\left(-\frac{\|r_a^i - r_b^i\|}{2\sigma}\right)$, where σ is a kernel parameter. To only consider the strong similarity, if the distance between the instances r_a and r_b is smaller than T (hyper-parameter), $\mathscr{K}(r_a^i, r_b^i)$ is set to 1, and 0 otherwise.

To capture the similarity among networks, the similarity between networks N_i and N_j is calculated as:

$$S_I(N_i, N_j) = \frac{\sum_{a=1}^{M}\sum_{b=1}^{M} w_a^i w_b^j \mathscr{K}(x_a^i, x_b^j)}{\sum_{M}^{n_i} w_a^i \sum_{b=1}^{M} w_b^j} \tag{7}$$

where $w_a^i = \frac{1}{\sum_{u=1}^{M} \mathscr{K}(x_a^i, x_u^i)}$ is associated with each brain region r_a^i in N_i with the q-th kernel function.

3.4. Training scheme

For overcoming these issues of optimization, we introduce three strategies to optimize the models of hi-GCN.

1. Two-step training. Both f-GCN and p-GCN are supervised with their own loss function independently. The optimization procedure is shown in Fig. 4(a).
2. Jointly training. We optimize f-GCN and p-GCN together with a single loss in hi-GCN as shown in Fig. 4(b).
3. Pre-training. We first optimize the node embedding features with the loss function in f-GCN, then the overall hi-GCN is further trained on the same data under the supervision with the loss in hi-GCN, which is based on the f-GCN with trained weights as shown in Fig. 4(c).

4. Experiment

In this section, we present several sets of comparative experiments both on ABIDE and ADNI data sets. Next, we briefly introduce these comparative experiments. The experimental settings and results are described in detail in the next subsections.

4.1. Databases and preprocessing

We apply our model on two large and challenging databases for

Table 1
The parameter settings of network training on ABIDE and ADNI database.

parameter name	parameters
Optimizer	Adaptive optimizer (Adam)
learning rate	0.001
dropout rate	0.5
batch size	900
Training iterations	160
l2 regularization	5.10^{-4}
L (Convolutional Layers for ABIDE & ADNI)	12 & 8
M(number of ROI)	116
P and $\widetilde{P}$ for ABIDE & ADNI	768 & 512
Threshold T in graph kernel for ABIDE & ADNI	0.6 & 0.75
Number of cluster H	7
Kernel parameter γ	3
Z in graph pooling	1

binary classification tasks. The ABIDE database (Autism Brain Imaging Data Exchange Database) investigates the neural basis of autism [34]. It aggregates data from different 17 acquisition sites and openly shares rs-fMRI and phenotypic data of 1112 subjects. We select the same set of 866 subjects used in Ref. [35], comprising 402 individuals with ASD and 464 healthy controls acquired at 20 different sites. We use the data from the Preprocessed Connectome Project [36] to discriminate individuals with Autism Spectrum Disorder from normal controls. The ADNI was launched in 2003 by the National Institute on Aging (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), private pharmaceutical companies and non-profit organizations, as a $60 million, 5-year public-private partnership. We focus on using rs-fMRI to discriminate individuals with Mild Cognitive Impairment (MCI) from individuals diagnosed with Alzheimer's Disease (AD). We select the same set of 133 subjects used in Ref. [35], comprising 99 individuals with MCI and 34 diagnosed with Alzheimer's Disease (AD).

4.2. Performance on hierarchical GCN

In this experiment we evaluated the effectiveness of hi-GCN on both the ADNI and ABIDE databases using a 10-fold stratified cross validation strategy. The parameter setting of our model is shown in Table 1. Using the above setting, we carried out comprehensive experiments to demonstrate the performance of the hi-GCN model for graph classification. The important hyperparameters in our proposed model are $T \in \{0.3, 0.45, 0.6, 0.47, 0.9\}$, $\gamma \in \{1, 2, 3, 4, 5\}$ and H (varying from 0 to 1). To select optimal parameters of our approach and all competing methods, we further perform a 10-fold nested cross validation procedure. In each trial, a nested 5-fold CV is conducted to tune the hyperparameters in the training data. To test whether the results of our method and those of each competing methodThe parameter settings of networks are statistically different, we perform the Student's t-test (with the significance level at 0.05) on the metric values achieved by our method and each competing method.

We evaluate our hi-GCN on the task of graph classification to answer the following three questions:

Q1: How does the jointly modeling two graphs in hi-GCN help improve the graph representation learning ability and the classification accuracy?
Q2: Which learning strategy is effective on the optimization of hi-GCN?
Q3: How do the important hyperparameters in hi-GCN affect the network performance?

We compare our hi-GCN with the connectivity features based method, Eigenpooling GCN [33] and Population GCN [27].

Network Based Feature is a linear classification using a ridge classifier. In the connectivity networks, there exist a large number of low level features (i.e., $\frac{M\times(M-1)}{2}$, where M is the total number of ROIs) are extracted from the network as network features for subsequent feature

Table 2
Performance comparison of various methods. Each experiment is run 10 times and the average graph classification performance (Accuracy, AUC, Sensitivity and Specificity) is reported. The best results are bold. The values marked by $*$ indicate that our method achieves significantly different results compared with the competing methods.

ABIDE	Methods	ACC	AUC	Specificity	Sensitivity
	network based feature	0.559±0.001*	0.602±0.002*	0.578±0.001*	0.537±0.001*
	Eigenpooling GCN	0.586±0.001*	0.655±0.003*	0.596±0.001*	0.574±0.001*
	Population GCN	0.635±0.002*	0.675±0.002*	0.651±0.002*	0.617±0.002*
	BrainNetCNN	0.651±0.003*	0.728±0.002*	0.656±0.003*	0.624±0.005*
	CC	0.607±0.002*	0.624±0.003*	0.587±0.001*	0.559±0.004*
	t-BNE	0.655±0.001*	0.708±0.001*	0.621±0.002*	0.608±0.002*
	Graph Boosting	0.646±0.002*	0.725±0.003*	0.639±0.003*	0.617±0.003*
	Ordinal Pattern	0.641±0.002*	0.729±0.002*	0.628±0.001*	0.611±0.002*
	Hi-GCN(two-step)	0.665±0.002*	0.721±0.003*	0.675±0.002*	0.653±0.003*
	Hi-GCN(pre-training)	0.671±0.003	0.743±0.003	0.681±0.003	0.659±0.003
	Hi-GCN(jointly learning)	**0.672±0.003**	**0.745±0.003**	**0.684±0.003**	**0.659±0.003**
ADNI	Methods	ACC	AUC	Specificity	Sensitivity
	network based feature	0.684±0.003*	0.691±0.002*	0.743±0.003*	0.512±0.003*
	Eigenpooling GCN	0.749±0.002*	0.711±0.004*	0.779±0.002*	0.662±0.004*
	Population GCN	0.737±0.002*	0.726±0.005*	0.763±0.002*	0.661±0.003*
	BrainNetCNN	0.728±0.002*	0.738±0.003*	0.767±0.002*	0.652±0.004*
	CC	0.701±0.004*	0.715±0.004*	0.738±0.005*	0.526±0.005*
	t-BNE	0.716±0.006*	0.722±0.006*	0.755±0.003*	0.663±0.003*
	Graph Boosting	0.731±0.005	0.726±0.004*	0.766±0.004*	0.671±0.004*
	Ordinal Pattern	0.729±0.003*	0.737±0.006*	0.769±0.004*	0.675±0.004*
	hi-GCN(two-step)	0.754±0.003	0.756±0.005*	**0.785±0.003**	0.664±0.004*
	hi-GCN(pre-training)	0.756±0.002	0.778±0.003*	**0.785±0.003**	0.672±0.003*
	hi-GCN(jointly learning)	**0.756±0.002**	**0.789±0.004**	0.777±0.002	**0.695±0.003**

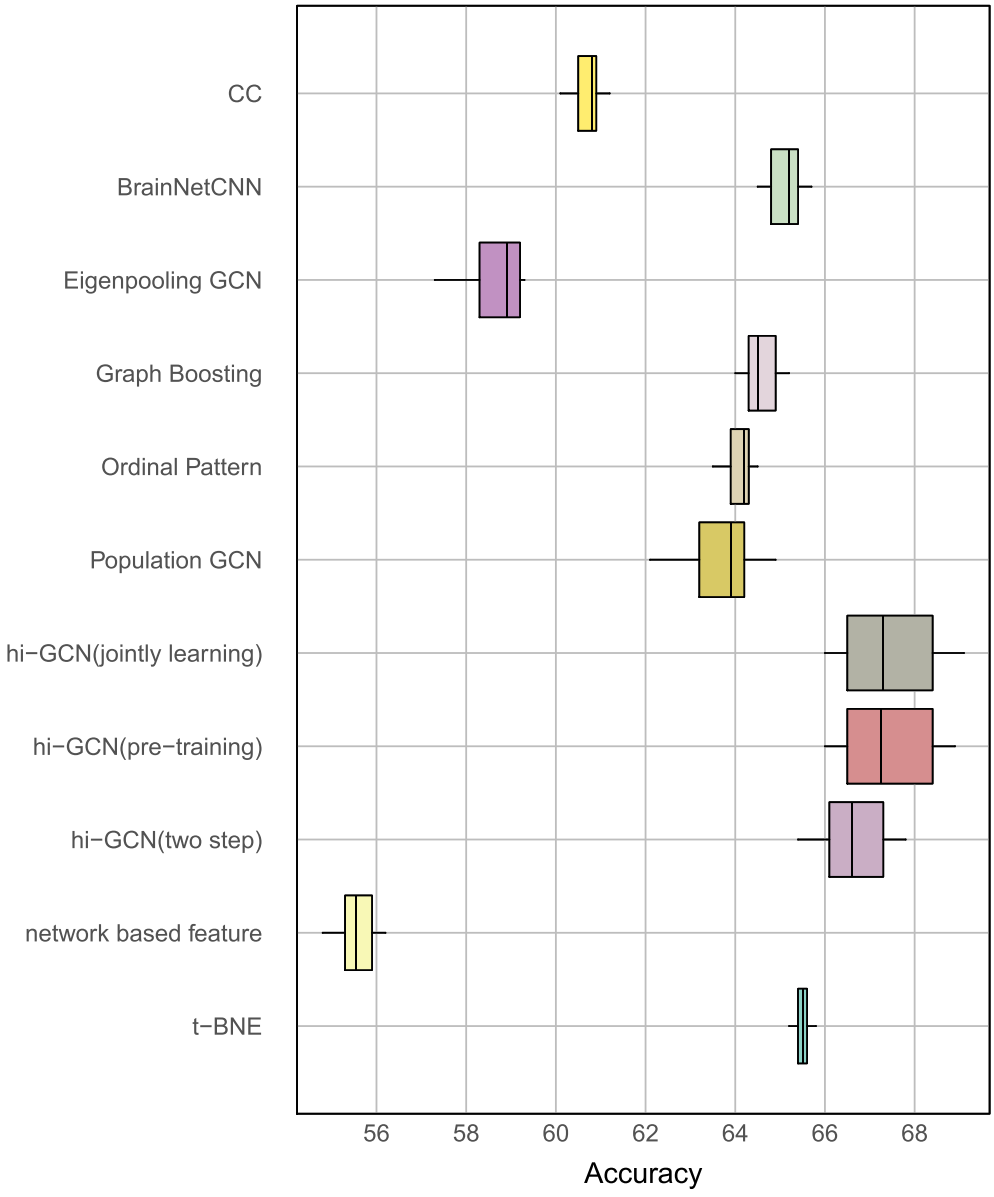


Fig. 5. The comparative boxplots on the ABDIE dataset.

selection and classification. Moreover, Recursive Feature Elimination (RFE) is used for feature selection. For fair comparison, the amount of features selected is equal to the dimension of the embedding vector generated by f-GCN.

Eigenpooling GCN [33] is an end-to-end trainable GCN with a pooling operator EigenPooling.

Population GCN [27] is a node level GCN for brain analysis in populations. The significant difference from the work in Ref. [27] are: 1) **node**: the embedding features of the nodes in the population is learned automatically rather than extracted; 2) **edge**: the similarity between nodes is calculated considering the structure of the brain functional network when constructing the population network. The apparent limitation of such model is that they can only learn on the vectorized node, which cannot effectively generalize the condition that the node is a graph describing the functional connectivity. The graph representation techniques have recently shifted from hand-crafted kernel methods to deep learning based end-to-end methods, which achieve better performance in graph-structured learning tasks. Moreover, the featured extracted prior to the classification may not be appropriate for GCN classification due to lacking the capacity of jointly learning.

BrainNetCNN [37]is composed of novel edge-to-edge, edge-to-node and node-to-graph convolutional filters that leverage the topological locality of structural brain networks.

Besides, we also compare hi-GCN with four state-of-the-art methods, including 2 topology-based representation approaches (i.e., Clustering Coefficient (CC) [38] and t-BNE [39]), and 2 subgraph-based representation approaches (i.e., Graph Boosting [40], and Ordinal Pattern [41]).

Clustering Coefficient (CC) [38] extracts local clustering coefficients as features of the brain network, by measuring the degree to which nodes in a network tend to cluster together (a measure that quantifies the cliquishness of the nodes). Then, Each network is converted into a feature vector to train a classifier.

Tensor-based Brain Network Embedding (t-BNE) [39] is a constrained tensor factorization model for brain network embedding. It can handle partially symmetric tensors, incorporates side information guidance and orthogonal constraint to obtain informative and distinct latent factors, and the obtained vectorized features are used for classification.

Graph Boosting (GB) [40] first mines subgraphs from each network

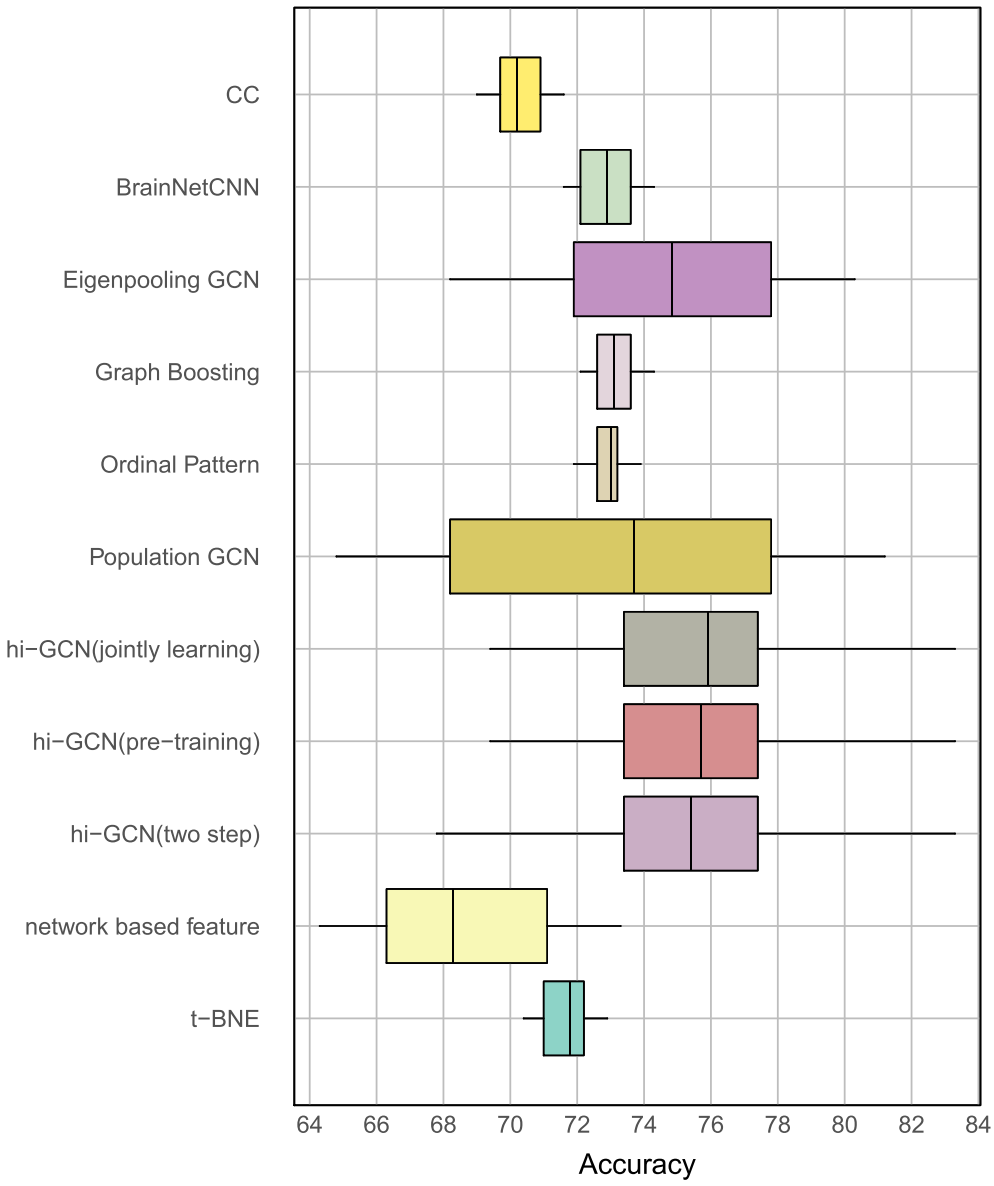


Fig. 6. The comparative boxplots on the ADNI dataset.

as features, and then learns subgraph-based decision stumps as weak learners. Finally, a Boosting algorithm in which subgraph-based decision stumps are used as weak learners is trained.

Ordinal Pattern (OP) [41] first mines frequent subgraphs via ordinal patterns that contains a sequence of weighted edges in brain connectivity networks of patients and normal controls. This method can simultaneously model both weight information (i.e., connectivity strength) and ordinal relationship of weighted edges in a brain connectivity network while relying on the subsequent classifiers.

Note that SVM with an RBF kernel is employed as the base classifier for the features extracted by both CC and OP. Both the regularization and the kernel parameters of SVM are tuned by 10 nested cross validation.

Experimental results are reported in Table 2 where the best results are boldfaced. Moreover, comparative boxplots across all folds between the comparable approaches are shown in Fig. 5 and Fig. 6 for both databases. As can be seen, our proposed method consistently showed the best performance over the baseline methods, and achieves statistically significant results on most of the results. These results reveal several interesting points:

1. For ABIDE, we obtain an average accuracy/AUC of 66.5%/72.1%, 67.1%/74.3% and 67.2%/74.5% in hi-GCN with different optimization scheme respectively, outperforming the other methods including network-based feature method(55.9%/60.2%), Eigenpooling GCN (58.6%/65.5%) and Population GCN (63.5%/67.5%). Results obtained for the ADNI database also show an increase in performance with respect to the competing methods. It demonstrates that our hi-GCN is effective for graph classification with brain disorders diagnosis regardless of the optimization.
2. The embedding learning methods with deep learning generally obtain better prediction results than the traditional extracted network connectivity features. Moreover, subgraph-based methods (i.e., GB and OP) generally obtain better performance than the topology-based methods, but their overall performance is not as good as hi-GCN's. It implies that considering the correlation among the instances helps promote the learning performance.
3. Previous studies typically utilize human-engineered features to represent brain connectivity networks, but these features may not be well coordinated with subsequent classifiers. It is worth noting that both topology-based and subgraph-based representation methods

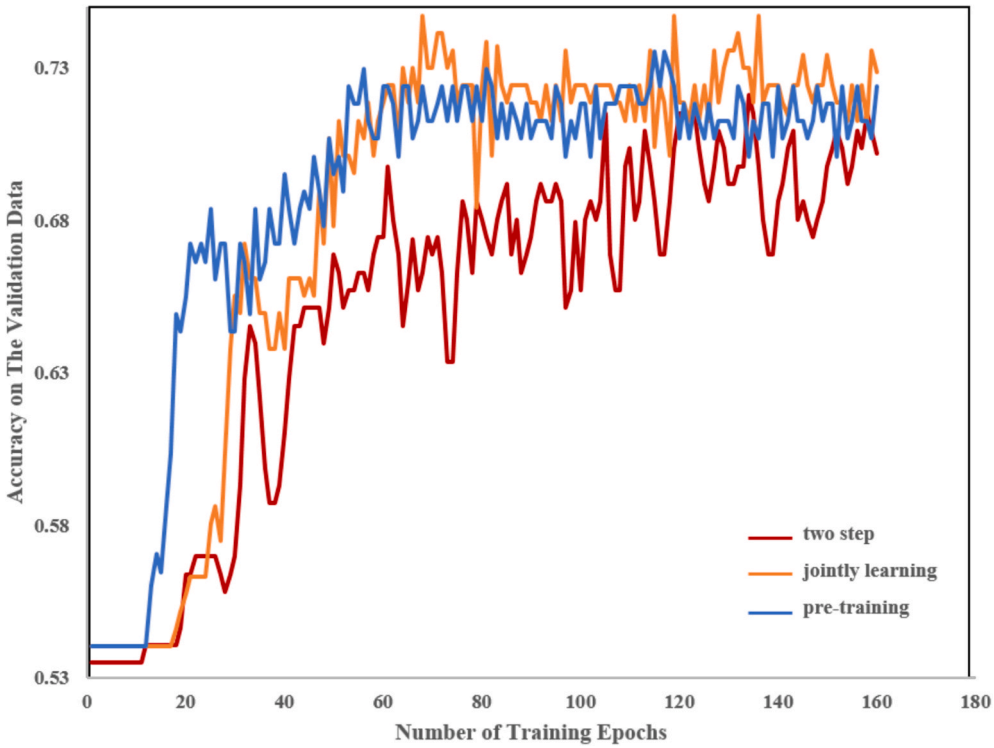


Fig. 7. Convergence behavior of three optimization methods for hi-GCN on ABDIE dataset.

usually first extract particular network representation from brain connectivity networks, and then reply on pre-defined classifiers for brain disease diagnosis. The feature extraction and classifier training are treated as two separate tasks in these methods, so potential inconsistency between features and classifiers may degrade the final performance of these methods. To address this issue, in our hi-GCN, the representation learning and classifier training are blended into a unified optimization problem, and the supervision information is introduced to the representation learning process, so that discriminative representations can be obtained.

4. Compared with Eigenpooling GCN and Population GCN, our hi-GCN performs the graph embedding learning from a hierarchical perspective considering the structure in individual brain network and the subject's correlation in the global population network, which can capture the most essential embedding features to improve the classification performance of disease diagnosis. The population network provides a powerful means for representing complex interactions between subjects. The correlation among instances provides rich information that can be leveraged to more comprehensively characterize the graph embedding. It demonstrates that analyzing the population graph can greatly benefit the graph embedding for GCN.
5. For the optimization in hi-GCN, the scheme of jointly training is more effective for the graph embedding learning. This experiment further demonstrates that optimizing the graph embedding with considering both the network structure and the neighborhood in the population simultaneously resulted in a better solution of graph embedding and a better prediction performance.

To evaluate the quality of the solution provided by our three different optimization algorithms, we compare them with respect to convergence behavior. According to Fig. 7, we can see both pre-training and jointly training converge faster than the two-steps. The pre-training method even starts convergence faster, but gets a slower convergence speed and almost equal accuracy compared to jointly training. Furthermore, it is also observed that the jointly training finally achieves better validation accuracy.

Table 3
Performance comparison of various methods based on imaging and non-imaging information. The best results are bold. The values marked by * indicate that our method achieves significantly different results compared with the competing methods.

ABIDE	Methods	ACC	AUC	Specificity	Sensitivity
	Population GCN	0.664±0.003*	0.723±0.004*	0.675±0.003*	0.652±0.004*
	hi-GCN(two-step)	0.719±0.002*	0.797±0.003*	0.732±0.002*	0.704±0.003*
	hi-GCN(pretraining)	0.728±0.003	0.817±0.004*	0.744±0.003	0.710±0.004*
	hi-GCN(jointly learning)	**0.731±0.003**	**0.823±0.004**	**0.746±0.003**	**0.714±0.003**
ADNI	Methods	ACC	AUC	Specificity	Sensitivity
	Population GCN	0.756±0.001*	0.813±0.002*	0.781±0.001*	0.683±0.002*
	hi-GCN(two-step)	0.779±0.002*	0.832±0.003*	0.798±0.002*	0.724±0.002*
	hi-GCN(pretraining)	0.782±0.002*	0.843±0.003*	0.803±0.002	0.720±0.003*
	hi-GCN(jointly learning)	**0.785±0.002**	**0.865±0.004**	**0.805±0.002**	**0.726±0.003**

Table 4
Performance comparison of various network similarity estimation in the construction of population network. The best results are bold. The values marked by * indicate that our method achieves significantly different results compared with the competing methods.

ABIDE	Methods	ACC	AUC
	metric Learning	0.721±0.004*	0.819±0.002
	Pdist	0.719±0.004*	0.811±0.003*
	graph kernel	**0.731±0.003**	**0.823±0.004**
ADNI	Methods	ACC	AUC
	metric Learning	0.772±0.003*	0.861±0.010
	Pdist	0.768±0.004*	0.847±0.011*
	graph kernel	**0.785±0.008**	**0.865±0.009**

4.3. Performance on different construction in population network

1) The effect of similarity estimation scheme with auxiliary information

Clinical and research studies commonly acquire complementary brain images for a more accurate and rigorous assessment of the disease status and likelihood of progression. To estimate the effectiveness of combining multi-modality, we follow the same work as in Ref. [27] by introducing the non-imaging complementary data (gender or acquisition site) in the estimation of subject's similarity.

For the ABIDE population graph, we choose subject's gender and acquisition site as the non-imaging modality; For the ADNI population graph, we choose age and gender of subjects. Let M be the non-imaging information of each subject. The similarity of non-imaging modality is calculated as:

$$S_{NI}(M_i, M_j) = \begin{cases} 1 & if \left|M_i - M_j\right| < \overline{T} \\ 0 & otherwise \end{cases}, \tag{8}$$

where $\overline{T}$ is a threshold value, and $\overline{T} = 2$ in our experiment.

Then, we integrate both imaging and non-imaging similarity together to be the integrated similarity score, which can be defined as follows:

$$S = \alpha S_I + (1-\alpha) S_{NI}, \tag{9}$$

where α is a parameter to adjust the contribution of the two similarity scores S_I and S_{NI} towards the integrated similarity score S. In our study, $\alpha = 0.5$.

From the results in Table 3, it is clear that the method with auxiliary information outperforms the methods using one single fMRI modality. This validates our assumption that the complementary information among different modalities is helpful for constructing the population network. For the experiments shown in Table 3, the hi-GCN(jointly learning) model gave the best results, and we used this model as our main learning model in the following experiments.

2) The effect of different similarity estimation scheme.

The definition of the graph's edges is critical in order to capture the underlying structure of the data and explain the similarities between the feature vectors. The influence of edge weight on the prediction performance was also investigated. In this section, we compare different similarity estimation methods of brain networks in the construction of the population network, involving graph kernel, metric learning and Pdist. All of them consider the multi-modality data during the similarity estimation according to Eq. (9). Metric learning is a method for learning a similarity metric between irregular graphs. In our work, we follow the work in Ref. [29] to choose siamese graph convolutional neural network (s-GCN) to learn a graph similarity metric in a supervised setting. Pdist computes the correlation distance between graph embedding vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ learned by the f-GCN. That is

$$\boldsymbol{K}_{pdist}(\boldsymbol{u}, v) = \frac{(\boldsymbol{u} - \overline{\boldsymbol{u}})\cdot(\boldsymbol{v} - \overline{\boldsymbol{v}})}{\|(\boldsymbol{v} - \overline{\boldsymbol{v}})\|_2 \|(\boldsymbol{v} - \overline{\boldsymbol{v}})\|_2}, \tag{10}$$

where $\overline{\boldsymbol{u}}$ and $\overline{\boldsymbol{v}}$ is the mean of the elements in vector.

At first glance at the results in Table 4, we can see that graph kernel consistently outperforms all other compared methods, which demonstrates the good behavior of the graph similarity considering the structure without embedding features compared with other related methods found in the literature. Most of the existing works focus on preserving network structures and properties in embedding vectors. However, some useful structural information may inevitably be lost. Direct similarity estimation with graph embedding is not appropriate. Different from graph kernel and Pdist, metric learning is trained to automatically predict the measure of similarity between the brain networks. In the metric learning, the label of matching (same class) or non-matching (different classes) is used to supervise Siamese graph convolutional neural network. However, it is difficult for Siamese graph convolutional neural network to learn the exact similarity for complex network structure with insufficient training data, which demonstrates that it may not be able to sufficiently capture underlying patterns by learning from network instances. The result indicates the similarity estimation of brain network is critical for the population network construction.

4.4. The influence of the hyperparameters of Hi-GCN

In the hi-GCN model, three important parameters are the number of clusters in the f-GCN (H), the threshold of graph kernel in constructing the population network (T) and the kernel parameter in the graph kernel (γ). In order to evaluate the impact of these parameters on the performance of hi-GCN, we conduct two experiments with various values for H, T and γ. Table 5 depicts the changes of accuracy as we vary the value of H, we can observe that the best classification performance is achieved when $H = 7$. It demonstrates that the hi-GCN with an appropriate cluster number can achieve optimal classification performance. Moreover, the optimal threshold T and γ of the graph kernel are unknown, and they play a vital role for the performance of graph similarity. We varied the value of the threshold $T \in \{0.3, 0.45, 0.6, 0.75, 0.9\}$ as well as $\gamma \in \{1, 2, 3, 4, 5\}$, and investigated the variation of performance with multiple values. When the threshold value T increases, the network becomes sparser for representing the most reliable correlation. Another important conclusion is that the effect of threshold T is more significant than γ. From Table 6 and Table 7, it can be seen that the best classification performance is achieved when $T = 0.6$ for ASD and $T = 0.75$ for AD. It implies that many edges are not helpful and reducing these edges does not influence the performance. The results further demonstrate that topological similarity between functional connectivity network with appropriate sparsity and kernel parameter is important for network embedding learning.

Table 5
The varying performance with the respect to cluster number H during the Eigenpooling. The best results are bold.

H	1	2	3	4	5	6	7	8	9	10
ABIDE database	0.602	0.658	0.699	0.722	0.730	0.729	**0.731**	0.731	0.708	0.711
ADNI database	0.557	0.657	0.663	0.687	0.729	0.783	**0.785**	0.782	0.784	0.742

Table 6
The varying performance with respect to T and γ during the estimation of node similarity with graph kernel. The best results are bold.

T/γ	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
$T = 0.3$	0.632±0.002	0.657±0.002	0.651±0.003	0.642±0.002	0.629±0.001
$T = 0.45$	0.689±0.002	0.696±0.003	0.698±0.002	0.687±0.003	0.679±0.002
$T = 0.6$	0.721±0.002	0.725±0.003	**0.731±0.003**	0.719±0.004	0.716±0.003
$T = 0.75$	0.671±0.002	0.698±0.003	0.703±0.003	0.711±0.002	0.702±0.002
$T = 0.9$	0.663±0.002	0.678±0.003	0.685±0.004	0.698±0.002	0.671±0.002

Table 7
The varying performance with respect to T and γ during the estimation of node similarity with graph kernel on ADNI database. The best results are bold.

T/γ	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
$T = 0.3$	0.632±0.004	0.621±0.004	0.602±0.005	0.594±0.004	0.572±0.003
$T = 0.45$	0.691±0.004	0.709±0.006	0.693±0.006	0.668±0.006	0.652±0.005
$T = 0.6$	0.749±0.006	0.762±0.007	0.763±0.008	0.754±0.006	0.752±0.006
$T = 0.75$	0.743±0.006	0.756±0.007	**0.785±0.008**	0.781±0.007	0.759±0.006
$T = 0.9$	0.719±0.004	0.743±0.005	0.752±0.006	0.763±0.006	0.726±0.004

Table 8
The comparison among different classifiers with previous methods for ASD vs. NC on ABIDE dataset.

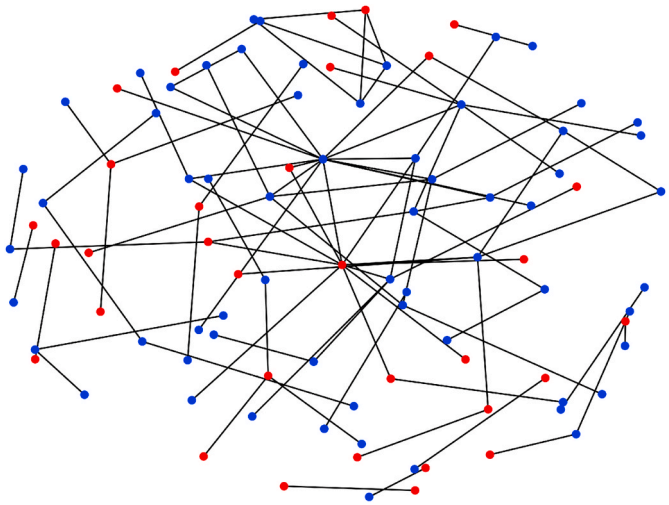
Method	Feature	Classifier	Data	CV	ACC
Sarah2018 [27]	Brain connectivity feature	GCN	403 ASD vs. 468 NC	10-CV	69.5%
Abraham2017 [42]	Tangent space embedding	l2-regularized classifiers (SVC-l2 and ridge classifier).	403 ASD vs. 468 NC	10-CV	67%
Dadi2019 [35]	Network connectivity feature	l2-regularized logistic regression	402 ASD vs. 464 NC	random 100-CV (75% for training, and 25% for test)	69.7%
Eslami 2019 [43]	No feature extraction	ASD-DiagNet (with augmentation)	505 ASD vs. 530 NC.	10-CV	69.2% (70.1%)
Heinsfeld 2018 [5]	No feature extraction	Two stacked denoising autoencoders	505 ASD vs. 530 NC	10-CV	70%
Our method	No feature extraction	hi-GCN(jointly learning)	402 ASD vs. 466 NC	10-CV	**73.1%**

4.5. Comparisons with prior works

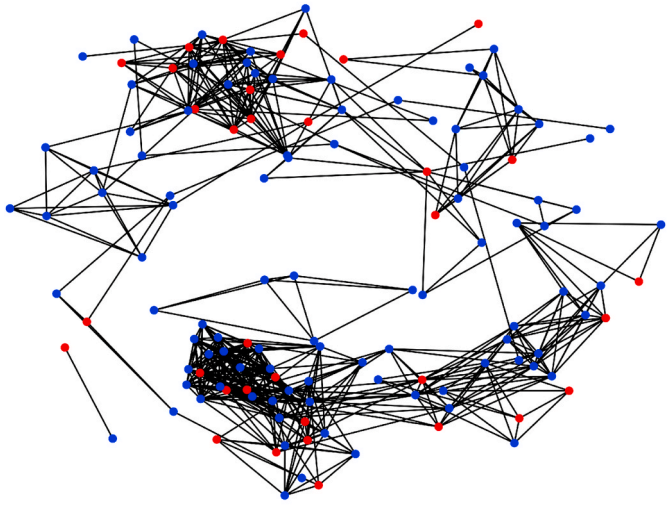
Table 8 compares the results of our ASD vs. NC classification with prior works in terms of accuracy as reported in the respective references. In general, two types of the methods are usually developed for the diagnosis of the disorder diseases: (1) the traditional machine learning procedure (feature extraction and classification) and (2) the deep learning procedure (an end-to-end procedure). For the traditional machine learning procedure, a straightforward solution that has been extensively explored is to first derive features from brain networks. Dadi [35] conducts a sufficient comparison (8 different ways of defining regions-either pre-defined or generated from the rest-fMRI data-3 measures to build functional connectomes from the extracted time-series and 10 classification models to compare functional interactions across subjects). Through the comparison, the optimal choices in functional connectivity prediction pipeline is brain regions defined with regions using DictLearn, connectivity matrices parametrized by their tangent-space representation, and an l2-regularized logistic regression as a classier. The best accuracy is 72.5% (median) and 84.5% (the 95th percentile) over cross-validation folds (n = 100) on the ADNI dataset with the same samples. On the ABIDE dataset, the best accuracy is 71.1% (median) and 75.6% (the 95th percentile). Abraham also employed the same strategy with the best pipeline and obtained a similar performance (accuracy is 67%) [42]. The most important limitation of the traditional machine learning procedure is that feature extraction and model learning are treated as two separate tasks in these methods, so potential inconsistency between human-engineered features and classifiers may degrade the final performance of these methods. Moreover, relying solely on subject-specific imaging feature vectors fails to model the interaction and similarity between subjects, which can reduce performance. Compared with the traditional machine learning procedure, the expressive power of deep learning to extract the underlying complex patterns from data has been well recognized. The power of deep learning lies in automatically learning relevant and powerful features for any perdition task, which is made possible through end-to-end architectures. Heinsfeld et al. [5] and Eslami et al. [43] have proposed a joint learning procedure using an autoencoder and a single layer or multi-layer perceptron which results in improved quality of extracted features and optimized parameters for the model. From Table 9, we can observe that our hi-GCN usually achieves competitive performance against both deep learning methods proposed in Ref. [5,43]. The underlying reason could be that 1) the brain network has complicated structures that contain rich underlying information, although an autoencoder network is used to learn a network embedding and classification in an unified fashion, they are applied on the extracted connectivity feature rather than the original

Table 9
The comparison among different classifiers with previous methods for AD vs. MCI on ADNI dataset.

Method	Feature	Classifier	Data	CV	ACC
Dadi2019 [35]	Network connectivity feature	l2-regularized logistic regression	40 AD vs. 96 MCI	random 100-CV (75% for training, and 25% for test)	72.2%
Our method	No feature extraction	hi-GCN (jointly learning)	34 AD vs. 99 MCI	10-CV	**78.5%**



(a) The ABDIE dataset



(b) The ADNI dataset

Fig. 8. The illustration of population network for two datasets.

network data; 2) they do not consider the contextual information in the population. Although this comparison is not done on the same proportion of the database, our hi-GCN exceeds previously published ABIDE findings. The best accuracy of previous works is 0.701, which is obtained by an autoencoder with data augmentation in Ref. [5]. Moreover, we find that the deep learning model with the end-to-end manner performs as well or worse than the traditional learning methods with connectivity feature extraction combined with ridge or SVM. Few brain network classification studies have explored the correlation among the brain regions and subjects, which are critical in neuroimaging research. To the best of our knowledge, the work of Parisot et al. [27] is currently relevant with ours for ASD diagnosis on the whole ABIDE dataset. Parisot et al. define a subject's feature vector as its vectorized functional connectivity matrix, and employ a ridge classifier to select the most discriminative features from the training set. With the selected connectivity information as the subject's feature, a population is represented as a graph where its vertices are associated with the extracted image-based feature vectors and the edges encode phenotypic information. However, the GCN model is trained on the traditional network features rather than on the original network. Experiments demonstrate that the proposed hi-GCN model performs better than the GCN model proposed in Ref. [27], indicating simultaneously taking both the brain region and subject correlations into account is important. We also compare our method with several recent state-of the-art methods reported in the literature using rs-fMRI data for AD vs. MCI classification. Few works for discriminating AD and MCI with functional brain network exist in the literature. The only comparable method is proposed in Ref. [35]. Experimental results on the ADNI database demonstrate that our method substantially outperforms the existing classification method for AD vs. MCI. Results obtained for the ADNI database show an increase in performance with respect to the competing methods in Table 9. Overall, the results demonstrate that hi-GCN can improve upon state-of-the-art algorithms not just in traditional classification methods, but also in deep learning methods.

4.6. Ablation study and discussion

To our knowledge, this is the first attempt to incorporate the region-to-region brain activity correlations and the subject interactions among population into a unified model for functional connectivity network analysis. In our work, the population network provides a powerful means for representing complex interactions between subjects. The key to our hi-GCN model is the subject's initial embedding and similarity among the subjects in the population network. In order to investigate the role of them in our model intuitively, we made sufficient ablation study by varying the threshold of the population network, exchanging the strategies of subjects' initial embedding and the similarity estimation, fusing the network embedding with the graph properties as node features, employing the traditional classification method, and employing other graph neural networks for the population network.

1) Varying the threshold of the population network

Network similarity is a quantitative measurement of topology and attribute characteristics between subjects in the population network. Different thresholds determine their corresponding different levels of topological structure in the population network. In other words, the thresholded connectivity networks with larger threshold often preserve fewer connections and thus are sparser in connections. Therefore, it is important to identify the optimal trade-off between the information gain by the removal of the noisy correlations and the loss due to removal of potentially useful weak correlations. Both the population networks of AD and ASD with the optimal T value are shown in Fig. 8.

We can see from Fig. 9 that when the T value increases, the performance of hi-GCN first goes up, and when T goes beyond 0.6 for ABIDE and 0.7 or 0.8 for ADNI, the performance (ACC and AUC scores) starts to decline. It demonstrates that more neighborhood information help learn better node embedding. Nevertheless, too much neighborhood inevitably leads to over-smoothing. If T is too large, nodes (subjects) of the population network cannot get sufficient information from correlated nodes (subjects). When $T = 1$, it is equivalent to the case without considering the correlation among the subjects.

2) Exchanging the strategies of subjects' initial feature and similarity estimation

To better evaluate and understand the effectiveness of the learned embedding by f-GCN and the similarity of the network structure of hi-GCN, we conduct the comparison by exchanging the strategies of subjects' initial embedding and the similarity estimation in this group of experiments, including (1) embedding (subjects' initial feature)+graph kernel (similarity), which is the chosen strategy in our hi-GCN; (2) embedding + learned embedding by f-GCN (similarity); (3) graph kernel (subjects' initial feature)+embedding(similarity), in which each network is represented by a feature vector corresponding to the similarity between this network with the other ones with the WL Graph Kernel (WLGK) [44]. The experimental results are reported in Table 10.

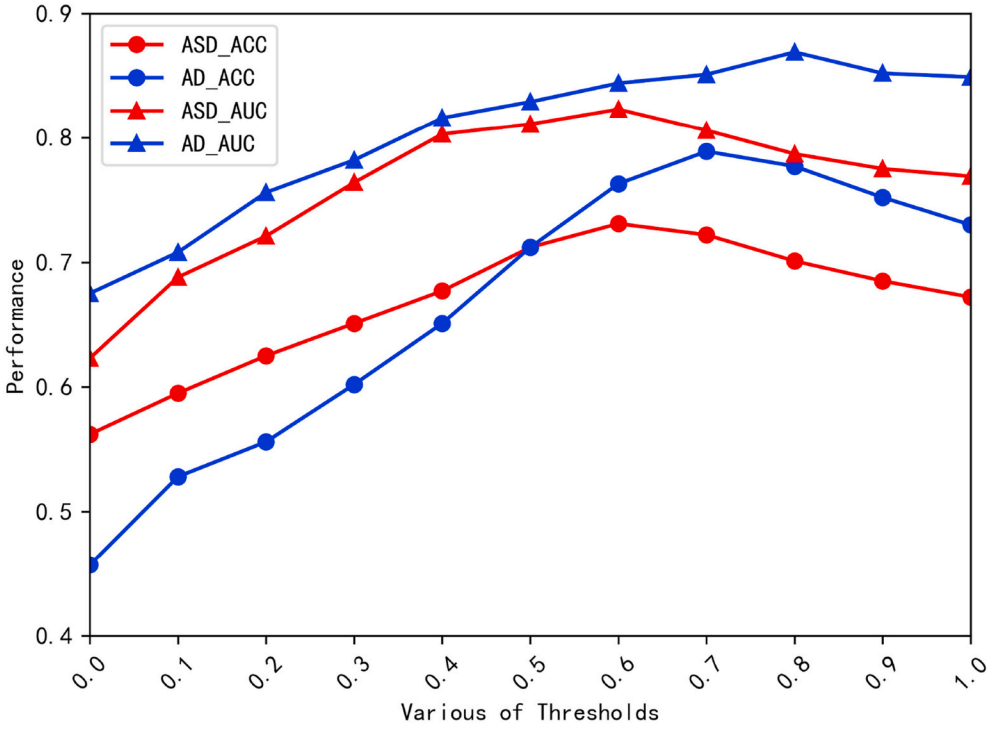


Fig. 9. Performance of different thresholds (T) for the population network.

Table 10
Performance comparison of various strategies of subjects' initial feature and similarity estimation. The best results are bold.

ABIDE			ADNI		
Methods	ACC	AUC	Methods	ACC	AUC
embedding + graph kernel	**0.731**	**0.823**	embedding + graph kernel	**0.785**	**0.865**
graph kernel + embedding	0.719	0.817	graph kernel + embedding	0.746	0.844
embedding + embedding	0.710	0.809	embedding + embedding	0.738	0.829

Table 11
Performance comparison of fusing the embedding with the various graph properties as node features. The best results are bold.

ABIDE			ADNI		
Methods	ACC	AUC	Methods	ACC	AUC
Only embedding	0.731	0.823	Only embedding	0.785	0.865
embedding + CC	0.728	0.823	embedding + CC	0.776	0.861
embedding + t-BNE	0.740	0.832	embedding + t-BNE	**0.789**	**0.872**
embedding + Ordinal Pattern	**0.756**	**0.857**	embedding + Ordinal Pattern	0.784	0.871

As anticipated, our strategy adopted in hi-GCN provides the best results.

3) Fusing the embedding with the graph properties as node features

In order to test whether the performance can be improved by combining with the graph properties as node features or not, the features describing the graph properties are extracted and incorporated into the node features in the population network. Besides the embedding obtained by f-GCN, we try to incorporate the topology-based representation or subgraph-based representation into the learned embedding. From Table 11, we can see that the proposed 'embedding + Ordinal Pattern' and 'embedding + t-BNE' generally improve the performance compared with the only embedding, which demonstrates that external information except CC is complementary to the learned embedding and can help improve the representation of the brain network. Integrating knowledge regarding the graph property could facilitate the learning of the network embedding.

Table 12
Performance comparison of embedding obtained by our hi-GCN and f-GCN for various classifiers. The best results are bold.

ABIDE	Methods	SVM	Random forest	Logistic regression
	embedding obtained by f-GCN	0.772	0.796	0.757
	embedding obtained by hi-GCN	**0.808**	**0.814**	**0.789**
ADNI	Methods	SVM	Random forest	Logistic regression
	embedding obtained by f-GCN	0.818	0.811	0.805
	embedding obtained by hi-GCN	**0.845**	**0.846**	**0.828**

4) Evaluating the embedding with the various traditional classifiers

In order to investigate the effectiveness of the learned feature representations, we compare the embedding obtained by our hi-GCN and f-GCN using various traditional classification models. For the regularization and the kernel parameters of SVM, the regularization parameter of logistic regression, and the number of trees of Random forest are tuned by a 10-fold nested cross validation. The result is shown in Table 12. Overall, the results suggest that embeddings obtained by our hi-GCN can enable more accurate capacity as compared to the ones by f-GCN for different classification models. It confirms our initial hypothesis that the joint learning from two levels enables us to learn a latent embedding such that both the structural and relational properties of the network can be encoded and preserved.

Table 13
Performance comparison of various methods with different version of GCN models. The best results are bold.

ABIDE			ADNI		
Methods	ACC	AUC	Methods	ACC	AUC
GCN	0.731	0.823	GCN	0.785	0.865
GAT	0.735	0.837	GAT	0.792	0.878
Higher-order GCN	**0.742**	**0.841**	Higher-order GCN	**0.796**	**0.882**

5) Evaluating the hierarchical embedding learning with various GCN models

The GCN model is employed in the population network, and it is a first-order model and no attention mechanism. Recently, attention mechanisms have been widely used in various tasks. The goal of the attention mechanism is to select information that is relatively critical to the current task from all input. We employ GAT (Graph Attention Networks) [45] with leveraging masked self-attentional layers to assign different weights to different nodes in a neighborhood. On the other hand, we perform higher-order convolutions by incorporating higher-order proximity via random walks in GCN. Integrating more inherent information into the proposed hierarchical embedding learning framework, which is expected to gain better results. Specifically, we employ a random walk sampling process on graphs to obtain the higher-order proximity representations. In our framework, the random walk strategy helps incorporate the higher-order structural information into the graph representations, which further allows for the higher-order graph convolutions to capture community and organizational structure of the population network. In Table 13, we report the comparison results of the different versions of GCN models. We can observe that both higher-order GCN and GAT perform better than the GCN model, indicating that more correlation or attention mechanisms applied in population network are more effective in capturing more information and learning more accurate brain network representations. The results further demonstrate the advantage of a hierarchical embedding learning model with two level GCN models. Furthermore, the results suggest that devising effective strategies to learn the network embedding in the population graph is essential to facilitate the learning of the brain network embedding and the diagnosis performance.

5. Conclusion

Recently, functional connectivity networks constructed from the functional magnetic resonance image (f-MRI) hold great promise for distinguishing the patients with neurological disorders from Normal controls. Network embedding is aimed at learning compact node representations based on network topology to facilitate the task of graph classification. In order to achieve a better graph embedding from brain networks, we develop a novel and principled framework for network embedding learning by efficiently integrating correlations among the subjects in a population with GCN. We conduct extensive experiments on real-world information networks to verify the effectiveness of our model, which demonstrates its superior performance compared with state-of-the-art baselines. It also achieves faster training and easier convergences.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, “Hi-GCN: A hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction”.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (No.62076059) and the Fundamental Research Funds for the Central Universities (No. N2016001).

References

[1] J. Cummings, G. Lee, A. Ritter, M. Sabbagh, K. Zhong, Alzheimer’s disease drug development pipeline: 2019, Alzheimer’s Dementia: Translational Research & Clinical Interventions 5 (2019) 272–293.
[2] E. Nichols, C.E. Szoeke, S.E. Vollset, N. Abbasi, F. Abd-Allah, J. Abdela, M.T. E. Aichour, R.O. Akinyemi, F. Alahdab, S.W. Asgedom, et al., Global, regional, and national burden of alzheimer’s disease and other dementias, 1990–2016: a systematic analysis for the global burden of disease study 2016, Lancet Neurol. 18 (1) (2019) 88–106.
[3] A. Association, 2019 alzheimer’s disease facts and figures, Alzheimer’s Dementia 15 (3) (2019) 321–387.
[4] Q. Li, X. Wu, L. Xu, K. Chen, L. Yao, R. Li, Multi-modal discriminative dictionary learning for alzheimer’s disease and mild cognitive impairment, Comput. Methods Progr. Biomed. 150 (2017) 1–8.
[5] A.S. Heinsfeld, A.R. Franco, R.C. Craddock, A. Buchweitz, F. Meneguzzi, Identification of autism spectrum disorder using deep learning and the abide dataset, Neuroimage: Clinical 17 (2018) 16–23.
[6] G.S. Bajestani, M. Behrooz, A.G. Khani, M. Nouri-Baygi, A. Mollaei, Diagnosis of autism spectrum disorder based on complex network features, Comput. Methods Progr. Biomed. 177 (2019) 277–283.
[7] M. Khosla, K. Jamison, G.H. Ngo, A. Kuceyeski, M.R. Sabuncu, Machine learning in resting-state fmri analysis, Magn. Reson. Imag. 64 (2019) 101–121.
[8] S. Qi, S. Meesters, K. Nicolay, B.M. ter Haar Romeny, P. Ossenblok, The influence of construction methodology on structural brain network measures: a review, J. Neurosci. Methods 253 (2015) 170–182.
[9] Y. Zhu, S. Qi, B. Zhang, D. He, Y. Teng, J. Hu, X. Wei, Connectome-based biomarkers predict subclinical depression and identify abnormal brain connections with the thalamus and lateral habenula, Front. Psychiatr. 10 (2019) 371.
[10] S. Qi, Q. Gao, J. Shen, Y. Teng, X. Xie, Y. Sun, J. Wu, Multiple frequency bands analysis of large scale intrinsic brain networks and its application in schizotypal personality disorder, Front. Comput. Neurosci. 12 (2018) 64.
[11] A. Khazaee, A. Ebrahimzadeh, A. Babajani-Feremi, Application of advanced machine learning methods on resting-state fmri network for identification of mild cognitive impairment and alzheimer’s disease, Brain imaging and behavior 10 (3) (2016) 799–817.
[12] W. Mier, D. Mier, Advantages in functional imaging of the brain, Front. Hum. Neurosci. 9 (2015) 249.
[13] R. Salvador, J. Suckling, M.R. Coleman, J.D. Pickard, D. Menon, E. Bullmore, Neurophysiological architecture of functional magnetic resonance images of human brain, Cerebr. Cortex 15 (9) (2005) 1332–1342.
[14] J. Wang, X. Zuo, Y. He, Graph-based network analysis of resting-state functional mri, Front. Syst. Neurosci. 4 (2010) 16.
[15] J.D. Medaglia, Graph theoretic analysis of resting state functional mr imaging, Neuroimaging Clinics 27 (4) (2017) 593–607.
[16] T.-E. Kam, H. Zhang, Z. Jiao, D. Shen, Deep learning of static and dynamic brain functional networks for early mci detection, IEEE Trans. Med. Imag..
[17] D. Wen, Z. Wei, Y. Zhou, G. Li, X. Zhang, W. Han, Deep learning methods to process fmri data and their application in the diagnosis of cognitive impairment: a brief overview and our opinion, Front. Neuroinf. 12 (2018) 23.
[18] X. Chen, H. Zhang, S.-W. Lee, D. Shen, A.D.N. Initiative, et al., Hierarchical high-order functional connectivity networks and selective feature fusion for mci classification, Neuroinformatics 15 (3) (2017) 271–284.
[19] H. Guo, L. Liu, J. Chen, Y. Xu, X. Jie, Alzheimer classification using a minimum spanning tree of high-order functional network on fmri dataset, Front. Neurosci. 11 (2017) 639.
[20] M.A. Ebrahimighahnavieh, S. Luo, R. Chiong, Deep learning to detect alzheimer’s disease from neuroimaging: a systematic literature review, Comput. Methods Progr. Biomed. 187 (2020) 105242.
[21] A.S. Lundervold, A. Lundervold, An overview of deep learning in medical imaging focusing on mri, Z. Med. Phys. 29 (2) (2019) 102–127.
[22] G. Litjens, T. Kooi, B.E. Bejnordi, A.A.A. Setio, F. Ciompi, M. Ghafoorian, J.A. Van Der Laak, B. Van Ginneken, C.I. Sánchez, A survey on deep learning in medical image analysis, Med. Image Anal. 42 (2017) 60–88.
[23] X. Yue, Z. Wang, J. Huang, S. Parthasarathy, S. Moosavinasab, Y. Huang, S.M. Lin, W. Zhang, P. Zhang, H. Sun, Graph embedding on biomedical networks: methods, applications and evaluations, Bioinformatics 36 (4) (2020) 1241–1251.
[24] A. Grover, J. Leskovec, node2vec: scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864.
[25] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: large-scale information network embedding, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 1067–1077.
[26] T. N. Kipf, M. Welling, Semi-supervised Classification with Graph Convolutional Networks, arXiv preprint arXiv:1609.02907.
[27] S. Parisot, S.I. Ktena, E. Ferrante, M. Lee, R.G. Moreno, B. Glocker, D. Rueckert, Spectral graph convolutions for population-based disease prediction, in:

International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2017, pp. 177–185.
[28] X. Li, N.C. Dvornek, Y. Zhou, J. Zhuang, P. Ventola, J.S. Duncan, Graph neural network for interpreting task-fmri biomarkers, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2019, pp. 485–493.
[29] S.I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, D. Rueckert, Metric learning with spectral graph convolutions on brain connectivity networks, Neuroimage 169 (2018) 431–442.
[30] S. Parisot, S.I. Ktena, E. Ferrante, M. Lee, R. Guerrero, B. Glocker, D. Rueckert, Disease prediction using graph convolutional networks: application to autism spectrum disorder and alzheimer's disease, Med. Image Anal. 48 (2018) 117–130.
[31] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, M. Joliot, Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain, Neuroimage 15 (1) (2002) 273–289.
[32] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems, 2016, pp. 3844–3852.
[33] Y. Ma, S. Wang, C.C. Aggarwal, J. Tang, Graph convolutional networks with eigenpooling, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 723–731.
[34] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F.X. Castellanos, K. Alaerts, J.S. Anderson, M. Assaf, S.Y. Bookheimer, M. Dapretto, et al., The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism, Mol. Psychiatr. 19 (6) (2014) 659–667.
[35] K. Dadi, M. Rahim, A. Abraham, D. Chyzhyk, M. Milham, B. Thirion, G. Varoquaux, A.D.N. Initiative, et al., Benchmarking functional connectome-based predictive models for resting-state fmri, Neuroimage 192 (2019) 115–134.
[36] C. Craddock, Y. Benhajali, C. Chu, F. Chouinard, A. Evans, A. Jakab, B. S. Khundrakpam, J. D. Lewis, Q. Li, M. Milham, et al., The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives, Neuroinformatics 41.
[37] J. Kawahara, C.J. Brown, S.P. Miller, B.G. Booth, V. Chau, R.E. Grunau, J. G. Zwicker, G. Hamarneh, Brainnetcnn: convolutional neural networks for brain networks; towards predicting neurodevelopment, Neuroimage 146 (2017) 1038–1049.
[38] C.-Y. Wee, P.-T. Yap, D. Zhang, K. Denny, J.N. Browndyke, G.G. Potter, K.A. Welsh-Bohmer, L. Wang, D. Shen, Identification of mci individuals using structural and functional connectivity networks, Neuroimage 59 (3) (2012) 2045–2056.
[39] B. Cao, L. He, X. Wei, M. Xing, P.S. Yu, H. Klumpp, A.D. Leow, t-bne, Tensor-based brain network embedding, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 189–197.
[40] T. Kudo, E. Maeda, Y. Matsumoto, An application of boosting to graph classification, in: Advances in Neural Information Processing Systems, 2005, pp. 729–736.
[41] D. Zhang, J. Huang, B. Jie, J. Du, L. Tu, M. Liu, Ordinal pattern: a new descriptor for brain connectivity networks, IEEE Trans. Med. Imag. 37 (7) (2018) 1711–1722.
[42] A. Abraham, M.P. Milham, A. Di Martino, R.C. Craddock, D. Samaras, B. Thirion, G. Varoquaux, Deriving reproducible biomarkers from multi-site resting-state data: an autism-based example, Neuroimage 147 (2017) 736–745.
[43] F. Saeed, T. Eslami, V. Mirjalili, A. Fong, A. Laird, Asd-diagnet, A hybrid learning approach for detection of autism spectrum disorder using fmri data, Front. Neuroinf. 13 (2019) 70.
[44] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, K. M. Borgwardt, Weisfeiler-lehman graph kernels., J. Mach. Learn. Res. 12 (9).
[45] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph Attention Networks, arXiv preprint arXiv:1710.10903.