MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

Representation Learning for the Graph Data

A thesis presented in partial fulfilment of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

Massey University, Albany, Auckland,

New Zealand

Jiangzhang Gan

2022

Abstract

Graph data consist of the association information between complex entities and also contain diverse vertex information. To make graph data analysis simple and effective, as the bridge between the original graph data and the graph application tasks, graph representation learning has become a hot research topic in recent years. Previous representation learning methods for the graph data may not reflect the intrinsic relationship between nodes due to the complexity of the graph data. Moreover, they do not preserve the topology of the graph data well, which will affect the effectiveness of the downstream tasks. To deal with these issues, the thesis studies effective graph representation learning methods in terms of graph construction and representation learning.

- We propose a traditional graph learning method under semi-supervised learning to explore parameter-free fusion of graph learning. Specifically, we first employ the Pearson correlation coefficient to obtain a fully connected Functional Connectivity brain Networks (FCN), and then to learn a sparsely connected FCN for every subject. Finally, the ℓ_1-SVM is employed to learn the important features and conduct disease diagnosis.
- We propose an end-to-end deep graph learning method under semi-supervised learning to improve the quality of initial graph. Specifically, the proposed method first extracts the common information and the complementary information among multiple local graphs to obtain a unified local graph, which is then fused with the global graph of the data to obtain the initial graph for the GCN model. As a result, the proposed method conducts the graph fusion process twice to simultaneously learn the low-dimensional space and the intrinsic graph structure of the data in a unified framework.
- We propose a multi-view unsupervised graph learning method. Specifically, the adaptive data augmentation first builds a feature graph from the feature space, and then designs a deep graph learning model on the original representation and the topology graph, respectively, to update the feature graph and the new representation. As a result, the adaptive data augmentation outputs multi-view information, which is fed into two GCNs to generate multi-view embedding features. Two kinds of contrastive losses are further designed on multi-view embedding features to explore the complementary information among the topology and feature graphs. Additionally, adaptive data augmentation and contrastive learning are embedded in a unified framework to form an end-to-end model.

All proposed methods are evaluated on real-world data sets. Experimental results demonstrate that our methods outperformed all comparison methods, compared to state-of-the-art methods.

Acknowledgements

Firstly, I would like to express my deepest gratitude to my advisor Dr. Xiaofeng Zhu for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. I am grateful for his time, commitment, and for every advice regarding my work and career. I will always keep these advice in my mind as I move forward. I will forever preserve thanks for that.

I also show my respect and sincere thanks to my co-supervisor Associate Professor Julian Jang-Jaccard for her support on my research, showing me the knowledge and invaluable guidance.

I would like to express my gratitude to my colleagues Dr. Rongyao Hu, Mr. Yonghua Zhu, Dr. Li Li, and Dr. Tong Liu for their companionship and assistance. I would like to thank all my collaborators that share my passion for this line of work, and I am looking forward to many more illuminated discussions in the future.

Lastly, I am grateful to my family for their love and support of my study. At the point of writing, I have been away from home for three years. I deeply thank my parents and my younger brother who give me endless support. Without their support, I would never have been able to commence my study in New Zealand.

Publications

- **J. Gan**, X. Zhu*, R. Hu, Y. Zhu, J. Ma, Z. Peng. Multi-graph fusion for functional neuroimaging biomarker detection [C]. The 29th International Joint Conference on Artificial Intelligence (IJCAI). 2020: 580-586.
- **J. Gan**, Z. Peng, X. Zhu*, R. Hu, J. Ma, G. Wu. Brain functional connectivity analysis based on multi-graph fusion [J]. Medical Image Analysis. 2021,71: 1-13.
- **J. Gan**, R. Hu, Y. Mo, Z. Kang, L. Peng, Y. Zhu, X. Zhu*. Multi-graph fusion for dynamic graph convolutional network [J]. IEEE Transactions on Neural Networks and Learning Systems. 2022.
- **J. Gan**, R. Hu, M. Zhan, Y. Mo, Y. Wan, X. Zhu*. Multi-view unsupervised graph representation learning [C]. The 31th International Joint Conference on Artificial Intelligence (IJCAI). 2022.
- R. Hu, **J. Gan**, X. Zhu*, T. Liu, X. Shi. Multi-task multi-modality SVM for early COVID-19 Diagnosis using chest CT data [J]. Information Processing & Management. 2022, 59(1).
- L. Peng, R. Hu, F. Kong, **J. Gan**, Y. Mo, X. Shi, X. Zhu*. Reverse Graph Learning for Graph Neural Network [J]. IEEE Transactions on Neural Networks and Learning Systems [J]. 2022.
- R. Hu, Z. Peng, X. Zhu*, **J. Gan**, Y. Zhu, J. Ma, G. Wu. Multi-band brain network analysis for functional neuroimaging biomarker identification [J]. IEEE Transactions on Medical Imaging. 2021, 40(12): 3843-3855.
- X. Zhu, B. Song, F. Shi, Y. Chen, R. Hu, **J. Gan**, W. Zhang, M. Li, L. Wang, Y. Gao, F. Shan, D. Shen*. Joint prediction and time estimation of COVID-19 developing severe symptoms using chest CT scan [J]. Medical Image Analysis. 2021, 67.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Machine learning is one of the core research areas of artificial intelligence. It aims to allow computers to automatically learn knowledge through designed algorithms [15, 39]. Recently, machine learning has gained great successes in diverse applications, *i.e.,* web search, intelligent surveillance, robotics, bio-medicine, geological exploration, and aerospace, due to the increase of the data scale and the advance of the computing resources [59]. Current machine learning usually consists of two processes, such as data representation and model learning [59]. Data representation refers to the transformation of input information into features that can be effectively used by computers. Model learning is constructed based on the data representation [9]. Thus, the success of machine learning depends on the data representation. Early machine learning methods used manual feature learning methods to first generate features from the original data and then apply them to model learning [69]. However, manual features are expensive and time-consuming. In recent years, representation learning is gradually becoming an important research area by learning discriminative representations.

Representation learning aims to capture semantic information of the data, so that it significantly improves the model effectiveness [81]. Representation learning emphasizes that data representations serve for the model learning, so a good representation needs to provide useful information to model learning [5]. From the data point of view, the data representation needs to extract valid information and reflects the true distribution of the data. From the algorithm point of view, the data representation needs to meet the format of the algorithm. Recently, representation learning has achieved excellent results in several fields. Traditional feature learning generally works by first designing some criteria and then selecting effective features based on these criteria [164]. Deep learning extracts the features of the data by building deep neural networks [185]. Compared with traditional representation learning methods, the neural network architecture used in deep learning has multiple network layers, which significantly improve the quality of representation/feature learning. However, in the past decades, the achievements of deep learning are mainly limited on

the study of Euclidean data and do not work well on the graph data (*e.g.,* Non-Euclidean data). With the prevalence of the graph data, researchers have begun to focus on representation learning for the graph data.

Graph data are widely used in real life [127], such as social media networks, academic citation networks, knowledge graphs and protein interaction graphs. Graph data not only consist of the association information between complex entities, but also contain diverse vertex information. The complexity and heterogeneity of the graph data bring more information to model learning, but they also bring new challenges to the analysis of the graph data. In order to make the analysis of the graph data simple and effective, as the bridge between the original graph data and the downstream tasks, graph representation learning has become a hot research topic. Specifically, graph representation learning can encode both node information and structure information of the graph to output discriminative representation. With the advancement of technology, current graph data present the characteristics of massive, heterogeneous, high-dimensional and multimodal, and machine learning have an increasing demand for graph representation learning method with high-performance.

1.2 Motivations

Representation learning for the graph data mainly includes two components, *i.e.,* graph construction and representation learning. Graph construction discovers the structural information between two nodes. The popular methods have k-Nearest Neighbor (kNN) method [190], the fully connected graph [181], and the ϵ-graph [32], *etc.* After obtaining the graph, representation learning represents the graph data as a set of low-dimensional vectors. Previous representation learning methods include traditional methods and deep learning methods. Traditional representation learning methods are simple and intuitive. Many traditional methods are designed to conduct dimensionality reduction, such as Locally Linear Embedding (LLE) [113], Laplacian Eigenmaps (LE) [8], and ISOMAP [7]. However, the performance of traditional methods may not be comparable to that of deep learning methods due to limitations in nonlinear representation capabilities. Deep learning methods transform the graph data into standard representations by making a series of strategies. As a result, the representation can be fed into neural networks for outputting semantic features.

Graph construction is the premise and foundation of graph representation learning. First, the quality of the graph directly affects the performance of graph representation learning [122]. In real applications, due to the influence of noise and outliers, the graph constructed from the original data often has wrong connections, which can degrade the robustness of the subsequent representation learning [17]. In traditional machine learning methods, a number of solutions have been proposed to remove or reduce the influence of noise and redundancy, including feature selection [76] and subspace learning [28]. However, current deep learning methods pay little attention to the issue, thus limiting the robustness of deep learning methods [70]. Second, in the process of the current graph representation learning methods, graph construction is separated from the subsequent tasks and is not updated during the model learning [112]. As a result, these

methods usually does not result in an optimal graph.

Previous graph representation learning methods mostly consider a single graph. For example, Jiang *et al.* generate a similarity matrix based on Euclidean distance [62]. Fu *et al.* construct an adjacency matrix based on Jaccard distance [34]. Due to the complexity of the graph data, the relationship between two data points can be described from different perspectives. Although multiple kinds of relationships make the connections between data points complex, they bring in more information for the representation learning. The studies of graph learning demonstrated the multi-graph structured data overcome the incompleteness problem of information [86]. Thus, it is of great significance to effectively integrate multi-graph information for downstream tasks. However, most of multi-graph learning methods do not consider the importance of different views in the process of graph fusion. This results in that the fused graph not only has more redundant information but also cannot guarantee the correctness of the structural relationship between nodes. For example, Tong *et al.* develop a graph fusion method for Alzheimer's disease diagnosis, which directly accumulates multiple graph structure information to generate the unified similarity matrix [138]. Some studies further integrate the multi-graph information to learn the accurate feature representation, but they cannot fully mine the complementary and common information of multiple graphs. Therefore, the multi-graph study is a challenging and promising topic in real applications.

- Parameter-free fusion of traditional graph learning. Different graphs result in different graph features. These features are usually heterogeneous and are in different feature spaces, and thus they are not comparable. Hence, it is straightforward to smooth all graph features so that they are homogeneous in the common feature space.
- Adaptive multi-graph leaning in graph convolutional networks (GCN). GCN outputs powerful representation by considering the structure information of the data to conduct representation learning, but its robustness is sensitive to the quality of both the feature matrix and the initial graph. However, in real applications, due to the noise and outliers, the initial graph often has inaccurate connections, which can degrade the robustness of the subsequent representation learning. For example, in biomolecule data, either the addition of edges or the removal of edges may change both the properties and the validity of compounds, thus reduces the performance of downstream models. Therefore, it is important for improving the quality of the initial graph to further improve the performance of the GCN models.
- Unsupervised multi-view graph representation learning. GCN has gained great popularity in tackling various tasks on the graph data. However, the recent studies raise concerns that the capability of the state-of-the-art GCNs in fusing node features is far away from optimal or even satisfactory. The weakness may severely hinder the capability of GCNs in classification tasks, since GCNs may not be able to adaptively learn deep correlation information between topological structures and node features. It remains challenging about how to extract the private and common embeddings from node features.

1.3 Contributions

This thesis aims to investigate different graph learning methods for different applications by overcoming the issues of previous graph learning methods. The main contributions of this thesis are summarized as follows:

- Objective 1 fuses the multi-scale features for functional neuroimaging biomarker identification by conducting joint graph feature learning and personalized disease diagnosis in semi-supervised learning.
- Objective 2 solves the issues of noise and redundancy in the original data by proposing a novel dynamic GCN method to jointly conduct graph learning and representation learning in a unified framework.
- Objective 3 enhances the capability of fusing topological structures and node features of GCN model by proposing two kinds of contrastive losses in unsupervised learning.

1.4 Thesis structures

In Chapter 2, we introduce the related work related to this thesis.

Chapters 3, 4 and 5 build the main contributions of this dissertation. In Chapter 3, we present a multi-graph fusion model for dealing with Resting state functional magnetic resonance imaging data in the brain functional connectivity analysis. In Chapter 4, we present an end-to-end graph learning model for dynamic multi-graph learning in semi-supervised node classification. In Chapter 5, we present unsupervised graph representation for dealing with feature graph and topology graphs in node classification.

Finally, Chapter 6 summarizes our contributions and outlines the future directions.

Chapter 2

Literature Review

In this chapter, we survey the research topics related to this thesis, including similarity measurements, graph representation learning, and multi-graph fusion.

2.1 Similarity measurements

Node similarity captures the relevance between two nodes in the graph data and has been recognized as an important research problem in various applications such as graph classification, node classification, link prediction, text mining and community detection [119, 124, 176]. In the literature, many similarity measurements were proposed and previous methods could be categorized into two types, *i.e.,* feature based similarity measurements and structure based similarity measurements.

2.1.1 Feature based similarity measurements

The popular methods of feature based similarity measurements include Manhattan distance [95], Euclidean distance [161], Pearson correlation coefficient [66], Cosine similarity [80].

Manhattan distance is calculated as the sum of the absolute differences between two vectors. Specifically, it is related to the L1 norm, *i.e.,* the sum absolute error:

$$sim_{Man}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{n} |a_i - b_i| \tag{2.1}$$

where sim_{Man} represents Manhattan distance between two vectors, *i.e.,* $\mathbf{a}$ and $\mathbf{b}$, and n is the number of dimensions.

Euclidean distance is the most common method of distance measurement. Its formulation is

$$sim_{Eui}(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{2.2}$$

where sim_{Eui} represents the similarity between node $\mathbf{a}$ and $\mathbf{b}$.

The Pearson correlation coefficient is the most common way of measuring a linear correlation. It measures the strength and direction of the relationship between two variables.

$$sim_{Pcc}(\mathbf{a},\mathbf{b}) = \frac{n\sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} a_i \sum_{i=1}^{n} b_i}{\sqrt{(n\sum_{i=1}^{n} a_i^2 - (\sum_{i=1}^{n} a_i)^2)(n\sum_{i=1}^{n} b_i^2 - (\sum_{i=1}^{n} b_i)^2)}} \tag{2.3}$$

Cosine similarity measures the similarity between two vectors in the inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.

$$sim_{Cos}(\mathbf{a},\mathbf{b}) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{2.4}$$

2.1.2 Structure based similarity measurements

The widely used methods of structure based similarity measurements have Jaccard distance [83], Path-based measurement [103], random walk [57].

Jaccard distance takes into consideration the intersection or overlap of features between two data points. The formulation is defined as:

$$sim_{Jac}(\mathbf{a},\mathbf{b}) = \frac{\mathcal{N}(\mathbf{a}) \cap \mathcal{N}(\mathbf{b})}{\mathcal{N}(\mathbf{a}) \cup \mathcal{N}(\mathbf{b})} \tag{2.5}$$

where $\mathcal{N}(\mathbf{a})$ represents the neighbors set of node $\mathbf{a}$.

Path-based measurement [22, 128] calculates the nodes' similarity in a network. This method considers the number of walks between nodes and walk length in the graph.

$$sim_{Pat} = \beta\mathbf{A} + \beta^2\mathbf{A}^2 + \beta^3\mathbf{A}^3 + ... = (\mathbf{I} - \beta\mathbf{A})^{-1} - \mathbf{I}. \tag{2.6}$$

where $\mathbf{A}$ is the adjacency matrix, $\beta < \frac{1}{\lambda_{max}}$, and λ_{max} is the maximum eigenvalue of $\mathbf{A}$. Since the time complexity, path based similarity indexes are difficult to be applied in the community detection. Most of path-based measurement methods only use the local information. Katz [189] and LHN-II [120] use global information of the graph, thus can overcome the shortage of the indexes using only the local information. However, the methods considering the global information are with high time complexity, *i.e.,* $O(N^3)$, which is not suitable for the graph with more than 1000 nodes.

An important family of similarity measurements are based on random walks, including Simrank [174], random walks with restarts [74], Personalized PageRank [6], and Deep Walk embeddings [16]. These methods capture both local and global graph structures, and thus they are widely used

in graph clustering, community detection, and many other applications. Random walk metric can be regards as a Markov chain with randomly selected nodes. Let $p_a(t)$ represent the probability of a random particle starting from node $\mathbf{a}$ and reaching other nodes after random walk t steps, we have

$$p_{\mathbf{a}}(t) = \mathbf{M}^T p_{\mathbf{a}}(t-1) \tag{2.7}$$

where $\mathbf{M} = (m_{ij})_{n\times n}$ represents the probability transition matrix obtained after the adjacency matrix is normalized. The random walk method iteratively applies Eq. (2.7) until the following termination conditions are satisfied.

$$\sum_{\mathbf{a}\in\mathbf{V}} (p_{\mathbf{a},\mathbf{b}}(t) - p_{\mathbf{a},\mathbf{b}}(t-1))^2 < \epsilon \tag{2.8}$$

where ϵ is a constant close to zero, $p_{\mathbf{a},\mathbf{b}}(t)$ represents the probability of node. Therefore, the similarity between node $\mathbf{a}$ and node $\mathbf{b}$ as follow.

$$sim_{Rw}(\mathbf{a},\mathbf{b}) = p_{\mathbf{a},\mathbf{b}}(t) \tag{2.9}$$

2.2 Graph representation learning

Previous methods of graph representation learning include traditional methods and deep methods.

2.2.1 Traditional graph representation learning

Traditional graph representation learning is a general term that constrains and optimizes the objective function to obtain a closed-form or numerical solution. Previous methods of traditional graph representation learning mainly include matrix factorization based methods and random walk based methods [49].

Matrix factorization based method has been widely used for the graph data analysis. These methods aim to decompose the data matrix into low-dimensional feature space while maintaining the hidden manifold structure and topological properties of the original data. The popular methods include principal component analysis (PCA) [1], locally linear embedding (LLE) [113], and SocDim [134]. PCA uses linear dimensionality reduction technique to obtain the graph representation. LLE first samples the input data and then fits the representation of the central node, followed by transforming the minimization loss function into the eigenvectors of the Laplacian matrix. In SocDim, the value of the node representation indicates that the node belongs to the corresponding strength of the community. Similar to LLE, the final optimization problem is also transformed into finding the eigenvectors for the normalized Laplace matrix. Although these method have been shown to be effective, performing matrix decomposition on matrices requires a large amount of memory and computational overhead. As a result, the computational efficiency of matrix decomposition methods is an urgent issue. In light of these limitations of existing methods, some researchers have started to focus on the extension of matrix factorization

based methods to large-scale graphs. For example, Qiu *et al.* leverage theories from spectral sparsification to efficiently sparsify the dense matrix, enabling significantly improved efficiency in representation learning [108], Wang *et al.* derive efficient update rules to learn the parameters of matrix factorization, and improve the efficiency of network embedding [151]. Furthermore, Gligorijevi *et al.* proposed a matrix factorization based method for multiplex network embedding, which successfully extends the matrix factorization based method to the field of complex network analysis [42].

In graph representation learning, it is popular to use random walk to capture structural relationships between nodes, *i.e.,* the 1st-order, 2nd-order, or high-order similarity between two nodes [67]. Compared to the adjacency matrix, the generation of random walk sequences only depends on the local information of the network, and thus is less complex in time and space. DeepWalk was the first method to learn node representations using random walk. Specifically, it generates several sequences of nodes and subsequently feeds the sequences into the Skip-Gram model to learn the representation of the nodes [106]. Node2vec further exploits a biased random walk strategy to capture the global structure information [46]. However, these methods only focus on preserving the network topology without taking into account the rich additional information of the network nodes, such as node labels, node attributes, and semantic descriptions of the nodes [25]. To deal with this issue, Li *et al.* present a discriminative deep random walk method that combines node labels in network data for graph representation learning [84]. Ahmed *et al.* propose a random walk based graph representation method. Specifically, it first maps the network nodes into a series of node types, and then performs a random walk of attributes to generate a sequence of node types, followed by using the Skip-gram model to learn the graph representation [4]. Recently, random walk methods have started to be combined with deep learning methods for graph representation learning.

2.2.2 Deep graph representation learning

The graph structure is highly nonlinear to node features, so it is essential to capture their nonlinear relationships for deep graph representation learning [140, 197]. Graph convolutional neural network (GCN) is the most classical method of deep graph representation learning by successfully defining convolutional operations on the graph data. Previous GCN methods for deep graph representation learinng can be grouped into spectral GCNs and spatial GCNs.

The spectral GCNs generate the data representation based on the spectral graph theory and the convolutional theorem [193]. To apply convolutional Neural Networks (CNN) on the graph data, Bruna *et al.* generalize convolutional kernals on the graph by performing point-wise product in the spectrum domain of the graph [12]. However, the disadvantage of this method is that it requires feature decomposition. As a result, the computational cost of feature decomposition is very high on large graphs, limiting its applications on the large scale graph with high-dimensional node features. To deal with this issue, Levie *et al.* propose a Cayley-Net to have linear time complexity [82]. Kipf *et al.* further improve the learning ability of GCNs by considering only the first-order neighborhood of nodes and increasing the perceptual field [75]. Simplifying graph convolution networks (SGCN) applies the k-th power of the graph convolution matrix in the neu-

ral network layers to capture the higher-order information in the graph and effectively reduces the computational effort [160]. Recent studies are focsed on improving spectral graph convolutional neural networks by exploring and replacing symmetric matrices. For example, adaptive graph convolutional network trains a residual graph to explore the residual substructure in the graph, and uses Mahalanobis distance to learn the optimal distance parameters between matrices [85]. Zhuang *et al.* propose a dual graph convolutional network to encode the local and global information of the graph data without stacking multiple convolutional layers [200]. Some studies were designed to reduce the complexity of the model by designing different graph pooling layers. Henaff *et al.* extend the convolution operator to high-dimensional data. Ying *et al.* propose a differentiable graph pooling method to reduce the input size of the graph layer [172].

In the spatial GCNs, the graph convolution layer is usually defined by using operations on the nearest neighbor samples of the nodes [184]. Therefore, spatial graph convolution methods focus on three key issues, *i.e.,* the selection of the central node, the size of the perceptual field, and the feature aggregation function. For the central node selection, PATCHY-SAN model organizes the central and neighbor nodes in an orderly manner [101]. The disadvantage of this method is that the node-centered metric function is not sufficiently determined, and the model might over-fitted in small-scale graph data. The perceptual flied size has been becoming a key parameter in the spatial graph convolution. Since the spatial graph convolution usually computes the neighbor nodes recursively, the size of the perceptual domain increases exponentially with the linear increase of the number of layers in the network increases. To reduce the complexity of the training process, it is necessary to design corresponding sampling methods. GraphSAGE generates the perceptual domain of the central node by multiple iterations [48]. Compared with traditional transductive graph representation learning methods, GraphSAGE learns a mapping function to generate an embedded representation for the nodes. Furthermore, Huang *et al.* propose a adaptive sampling method in which the sampler adaptively adjusts according to the variance reduction during training [56]. For the neighbor node feature aggregation, the popular approach is to perform linear or nonlinear transformations on the neighbor nodes. The popular aggregator strategies have mean aggregator, LSTM aggregator, and pooling aggregator. However, these methods only focus on the aggregation of first-order neighbors. Chen *et al.* propose a CoN-GCN to aggregate both the first-order and the second-order neighbor of the nodes on the graph data [19]. Thus, CoN- GCN is able to learn an efficient representation of the node.

2.3 Multi-graph fusion

The graph data may come from different domains, representations, views or modalities [52]. A natural question is to combine these information for improving the effectiveness of downstream tasks. The related technique is often referred to as multi-graph fusion. For a specific knowledge mining task, each graph has its own unique properties, but different graphs often contain complementary information that needs to be mined. The purpose of multi-graph fusion is to improve the generalization ability by jointly optimizing all graphs to effectively fusing information from different views. Previous multi-graph fusion methods can be divided into structure-level fusion

methods and feature-level fusion methods.

2.3.1 Structure-level fusion methods

Structure-level fusion methods focus on learning a common graph (*e.g.,* the fused graph) for downstream tasks [123]. The most straightforward method is the direct cumulative processing of multiple graph information. Specifically, it generates a new adjacency weight matrix by directly summing the matrix of weights of multiple matrices. For example, Wang *et al.* develop a low-rank based multi-view spectral clustering method. Tong *et al.* develop a graph fusion method for classification of Alzheimer's Disease, which directly accumulates multiple graph structure information to generate the unified similarity matrix [138]. These multi-graph fusion strategy treats all views equally, which might results in inferior performance. To improve the quality of the fused graph, many structure-level fusion methods perform graph learning and graph fusion in a unified framework, which effectively explores the complementary information of multi-graph data and potentially describe local geometric structures. For example, Wang *et al.* develop a self-attention network to integrate the node attributes with topological features to balance the disparity between the graph structure and node attributes for each similarity matrix [153]. Tang *et al.* propose a multi-graph clustering method that uses multiple similarity matrices as input [135]. Zhan *et al.* propose multi-view consistent graph clustering to learn a consistent graph [177]. These methods take into account the complementary and common information of different graphs, and therefore can output more robust node representations.

2.3.2 Feature-level fusion methods

Feature-level fusion method first extracts the low-dimensional feature vectors of each graph individually, and then uses fusion mechanisms to integrate the low-dimensional features of each graph, followed by feeding the fused features into the final decision model to derive the final decision results [77]. At present, the popular feature-level fusion methods have feature connected method and weighted fusion method.

Feature connected method first represents the multi-graph data in a uniform form, and then stitches it into a larger feature representation. For example, Jia *et al.* connects the spatial feature and the temporal feature of dynamic functional connectivity network for the sleep stage classification [60]. Kazi *et al.* first connects the features of different modalities and then uses them for personalized disease prediction [73]. Ding *et al.* stitches together the multi-scale features with pixel-wise local features into a new feature for HSI classification [29]. The advantage of feature connected fusion is that it achieves information compression and fast information processing. However, these methods only consider the association information between multiple graphs, but are not consider the difference between them. As a result, they cannot effectively describe the rich association information of multiple graph data. Recently, Kang *et al.* first fuse multi-hop heterogeneous features via discrete cross-correlations, and further fuse representations from two graphs by the attention mechanism [178].

The weighted fusion method focuses on learning the importance of different graphs by assigning

different weights to each graph. For example, Yao *et al.* fuse brain network features of different templates by the weighted fusion for the diagnosis of Alzheimer's disease [170]. He *et al.* propose a multi-graph convolutional-recurrent Neural Network to conduct five graph convolution operators [51]. Compared with structure-level fusion methods, feature-level fusion methods avoid early cross-contamination caused by noise and irrelevant information contained in each graph, and thus they retain the representation ability of different graph structures. The disadvantage of feature-level fusion methods is that more information will be lost in the learning process. Moreover, its model complexity is more complex than the structure-level fusion method.

Chapter 3

Multi-graph Fusion for Brain Functional Connectivity Analysis

3.1 Introduction

Resting state functional magnetic resonance imaging (rs-fMRI) is a non-invasive method to measure the spontaneous nerve activity of human brain in resting state, which provides a powerful tool for the study of brain function after nerve injury [10]. Using rs-fMRI data to construct brain functional connectivity network (FCN) can not only reveal the pathological basis of brain disease, but also help to develop biomarkers that quantify the reorganization mechanism of brain disease [45]. In recent years, brain network analysis methods based on rs-fMRI have been widely used in computer-aided diagnosis of various brain diseases [201].

Generally, the method of disease diagnosis based on rs-fMRI mainly includes three steps, *i.e.,* FCN construction, feature learning, and disease diagnosis. FCN model the functional association patterns between brain regions as networks, in which nodes correspond to brain regions of interest (ROIs) and edges represent functional connections between brain regions. At present, popular methods of constructing FCN have correlation-based approaches [21], Granger causality analysis [118], and regularized inverse covariance estimation [11]. Compared with other methods, the existing literature shows that correlation-based approaches can provide relatively high sensitivity to network connection detection [116]. Feature learning extracting meaningful features (*i.e.,* clustering coefficient, connection strength) from the FCNs, followed by feature selection to select the most discriminative feature subset for classification. Two simple feature selection methods have been widely applied currently, *i.e.,* t-test [2] and Lasso methods [47]. Disease diagnosis usually uses traditional machine learning methods (*i.e.,* SVM) to identify healthy subjects and patients [114]. At present, most studies based on brain functional connectivity mainly focus on the FCN construction and feature learning to improve the performance of disease diagnosis. However, due to the influence of noisy, individual variability, and inter-group heterogeneity, *etc.* there are many issues to be solved in brain network analysis.

On the one hand, FCNs constructed from rs-fMRI data plays a vital role in the diagnosis of brain neurological diseases. Due to different factors such as noises and the curse, it is a challenged task for constructing a FCN that can accurately reflect the functional connectivity of brain. First of all, the full-connected FCNs constructed by correlation-based methods usually have noise or false connection which cannot be explained and will affect the classification performance. To address these issues, existing methods (*i.e.,* [96, 169]) fine tune full-connected FCN by threshold method or k-NN based method to reduce the noise connection and network complexity. However, this method ignores that the number of connections may be different in different brain regions because of the individual heterogeneity, so it is unreasonable to use the same parameter (*i.e.,* threshold, k) for all subjects. Second, the brain is the most complex system in the human body, a single FCN may not be able to capture the subtle disruption of brain functional tissues caused by neurological diseases, because each network can only capture a part of the differences between groups [55]. Some methods (*i.e.,* [155, 158]) explored the construction of multiple networks based on rs-fMRI disease classification, but these methods did not take into account the joint optimize information from multiple FCNs and have some limitations. Third, the independent process for FCN construction ignores to considering the group effect so that the outputted feature representation has limited discriminative ability [187].

On the other hand, many fMRI-based methods extract functional features from FCNs to represent each subject, and input these features into a predefined classification model for disease diagnosis. However, these features are usually high-dimensional, so some methods use feature selection to select the most recognizable features to train the classification model. For example, Wee *et al.* extract local clustering coefficient features from both structural and functional connectivity networks, followed by t-test based feature selection for MCI identification [156]. Liu *et al.* extract connectivity strength features from FCNs, and perform feature selection using F-scores to identify patients with social anxiety disorder [89]. These studies show that feature selection can not only improve the diagnostic ability of models, but also help to discover biomarkers based on neural images. However, these methods regard feature selection and classification as independent tasks. The heterogeneity between the classifier and selected features may cause the selected features to be unsuitable for the classification task, thereby affecting the performance of the classification model. Although there are some methods (*i.e.,* [94, 149]) that propose to jointly perform feature learning and classification, the features extracted by such methods usually lack interpretability.

To solve the above issues, the chapter proposes a classification framework based on multi-graph fusion to accurately identify patients with brain neuro-diseases. The key of our method is to use a new fusion strategy to explore the different hierarchical structure of FCNs. Specifically, with the k-NN based method, we can construct multiple FCNs (*i.e.,* full-connected FCN and 1-NN FCN). In order to effectively joint optimize the information from multiple FCNs, we proposed a new multi-graph model to integrate these FCNs to enhance the common intrinsic structure and limit the error rate caused by the different structure. In the process of fusion, the number of edges of each node is updated iteratively, so that the number of neighbors of each node is learned automatically. After getting the fused FCN of each subject, we extract the upper triangle of the FCNs as the feature of each subject, and then use L1SVM to joint conduct feature selection

(*i.e.,* brain region selection) and classification (*i.e.,* disease diagnosis). The overview of our method is illustrated in Figure 3.1.

The major contributions of this work are three-fold. First, we develop a functional connectivity network based classification model for brain disease diagnosis, with feature selection and model training incorporated into a unified framework to solve the issues of heterogeneity between selected features and classifiers. Specifically, we use L1SVM to output the discriminative functional connectivity features and disease diagnosis results simultaneously. Second, we proposed a multi-graph fusion model, which fuse different network features and automatically learns the connections of various brain regions. In addition, our fusion model designs two regularization terms to achieve the group effect (*i.e.,* the homogeneity among samples from the same class and the heterogeneity among samples from different classes), which requires the subjects from the same class are close to each other and the subjects from different class to be far away from each other. As a result, our method solves the issue of individual heterogeneity. Third, our work outperforms four comparison methods in term of classification and feature selection, demonstrating promising performance in real application.

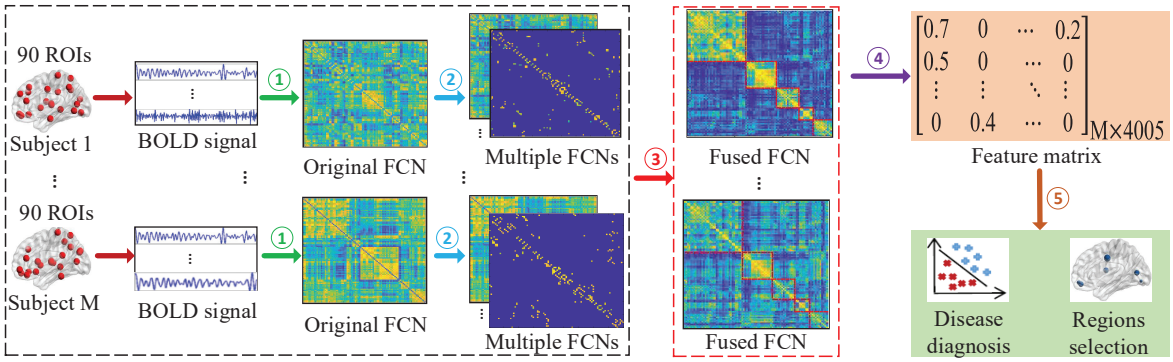


Figure 3.1: Framework of functional connectivity analysis based on multi-graph fusion. (1) rs-fMRI image pre-processing and original FCNs construction, (2) multiple FCNs construction , (1) and (2) can be finished offline, (3) multi-graph fusion by our proposed method, (4) feature exaction (*i.e.,* connection strength), (5) new feature matrix generation, (6) L1SVM based joint disease diagnosis and regions selection.

3.2 Method

Given the BOLD signal of the m-th subject among M subjects $\mathbf{B}^m \in \mathbb{R}^{n\times t}$ $(m = 1, ..., M)$ where n and t, respectively, represent the number of brain regions and the length of signals, in this paper, we first obtain multiple graphs (*i.e.,* FCNs) $\mathbf{A}^{m,v} \in \mathbb{R}^{n\times n}$ $(v = 1, ..., V)$[1] by the Pearson correlation analysis where V is the graph number, and then propose to learn a sparse-connected FCN (sparse FCN for short) $\mathbf{S}^m \in \mathbb{R}^{n\times n}$ for each subject so that it could automatically

[1]If the value of $\mathbf{A}^{m,v}$ is negative, we take its absolute value, *i.e.,* $a^{m,v}_{i,j} = |a^{m,v}_{i,j}|$.

learn the connection number of every node as well as is homogenous and discriminative to other sparse FCNs $\mathbf{S}^{m'}$ $(m \neq m')$.

3.2.1 Multi-graph fusion

Previous studies demonstrated that the sparse FCN is preferred in the representation learning of brain functional connectivity analysis ([72, 182]), compared to the fully-connected FCN, due to that 1) the fully-connected FCN lacks the interpretability; 2) the connectivity between two nodes may contain noisy connectivity (*i.e.,* either irrelevant or spurious connectivity) to affect brain functional connectivity analysis [79, 111]; and 3) neurologically, a brain region predominantly interacts only with a part of brain regions. Existing methods of FCN analysis usually obtain sparse FCNs from the fully-connected FCNs. Specifically, they design different techniques to learn sparse FCNs based on the fully-connected FCNs, such as sparse learning [31, 182] and clustering [156, 186]. However, these methods have limitations in the brain FCN analysis. First, existing methods usually assume that each node connects a fixed number of nodes. That is, the connection number is unchanged for all nodes. To achieve this, the sparse *k*-nearest neighbor (*k*NN) graph is constructed so that each node connects with *k* nodes. Such an assumption obviously ignores the fact that a brain region predominantly interacts only with a part of brain regions. Second, previous methods generate the sparse FCN of a subject independently from other subjects. On the one hand, by considering the heterogeneity across subjects, the FCNs obtained from these heterogeneous subjects possibly have different distributions. As a result, the robustness of the classifier constructed by these sparse FCNs will be affected. On the other hand, the independent process of representation learning makes it difficult to consider the group effect, *e.g.,* the discriminative ability across classes or subjects.

Given the fully-connected FCN connecting each node with all nodes, we obtain an extreme sparse FCN, *i.e.,* 1NN graph (excluding itself). By this way, we could obtain multiple graphs for each subject to solve the first issue of existing functional connectivity analysis. In this paper, we only use 2 graphs for every subject, *i.e.,* a fully-connected FCN and an extremely sparse FCN, by taking the following observations into account. The fully-connected FCN contains all connectivity information (*i.e.,* the most complex connectivity) and the extremely sparse FCN contain the least information (*i.e.,* the simplest connectivity). We expect to obtain a flexible connection number for every node based on the data distribution in the range $[1, n]$ where n is the node number. To do this, we design the following objective function to automatically learn specific connection number for the *m*-th subject $\mathbf{S}^m$ by fusing the information from multiple graphs.

$$\begin{aligned} &\min_{\mathbf{S}^m} \sum_{v=1}^{V} ||\mathbf{S}^m - \mathbf{A}^{m,v}||_F^2 \\ &s.t., \forall i, \mathbf{s}_{i,\cdot}^m \mathbf{1} = 1, s_{i,i}^m = 0, s_{i,j}^m \geq 0 \; if \; j \in \mathcal{N}(i), otherwise \; 0. \end{aligned} \tag{3.1}$$

where $\|\cdot\|_F$ indicates the Frobenius norm. $\mathbf{s}_{i,\cdot}^m$ and $s_{i,j}^m$, respectively, represent the i-th row of $\mathbf{S}^m$ and the element in the i-th row and the j-th column of $\mathbf{S}^m$. $\mathbf{1}$ and $\mathcal{N}(i)$, respectively, indicate the all-one-element vector and the set of nearest neighbors of the i-th node. After optimizing $\mathbf{s}_{i,\cdot}^m$ by

our proposed optimization method in Appendix, we obtain different non-zero numbers for every row, *i.e.,* $\mathbf{s}^m_{i,\cdot}$ in $\mathbf{S}^m$. This indicates that different nodes have different connection numbers for every subject.

Eq. (3.1) employs multiple graphs to conduct the feature or representation learning, aiming at selecting an optimal connection number between 1 and n. However, the optimization of $\mathbf{S}^m$ is independent on the optimization of $\mathbf{S}^{m'}$ $(m \neq m')$, which explores the inter-subject variability, but does not touch the issue of the heterogeneity across subjects. To address this issue, we propose the following objective function.

$$\begin{aligned}\min_{\mathbf{S}^1,...,\mathbf{S}^M,\mathbf{H},\mathbf{G}} & \sum_{m=1}^{M}\sum_{v=1}^{V}||\mathbf{S}^m-\mathbf{A}^{m,v}||_F^2+\alpha\mathcal{R}_1(\mathbf{H},\mathbf{G},\mathbf{S}^1,...,\mathbf{S}^M)\\ &+\beta\mathcal{R}_2(\mathbf{S^1},...,\mathbf{S}^M)\\ s.t., \forall i, \mathbf{h}_{i,\cdot}\mathbf{1}=1, h_{i,i}=0, h_{i,j}\geq 0\, if\ j\in\mathcal{N}(i), otherwise\ 0,\\ \mathbf{g}_{i,\cdot}\mathbf{1}=1, g_{i,i}=0, g_{i,j}\geq 0\, if\ j\in\mathcal{N}(i), otherwise\ 0,\\ \mathbf{s}^m_{i,\cdot}\mathbf{1}=1, s^m_{i,i}=0, s^m_{i,j}\geq 0\, if\ j\in\mathcal{N}(i), otherwise\ 0.\end{aligned} \tag{3.2}$$

where $\mathbf{H}\in\mathbb{R}^{n\times n}$ and $\mathbf{G}\in\mathbb{R}^{n\times n}$ are two variables, $\mathcal{R}_1(\mathbf{H},\mathbf{G},\mathbf{S}^1,...,\mathbf{S}^M)$ and $\mathcal{R}_2(\mathbf{S^1},...,\mathbf{S}^M)$ are regularization terms. We use the summation operator in the first term of Eq. (3.2) to learn the representations of all subjects in a unified framework, and design two regularization terms to achieve the group effect, *e.g.,* discriminative ability across subjects. We list the details of two regularization terms as follows.

First, we expect that positive subjects are similar or close to the positive template $\mathbf{G}$ while negative subjects are similar to the negative template $\mathbf{H}$. Hence, the subjects within the same class are close. It is noteworthy that $\mathbf{G}$ and $\mathbf{H}$, respectively, can be regarded as the common information of the positive class and the negative class. Moreover, the outputted templates could be widely applied in medical imaging analysis, such as guiding parcellations for new subjects and measuring the group difference [111]. To achieve this, we design $\mathcal{R}_1(\mathbf{H},\mathbf{G},\mathbf{S}^1,...,\mathbf{S}^M)$ as follows

$$\mathcal{R}_1(\mathbf{H},\mathbf{G},\mathbf{S}^1,...,\mathbf{S}^M)=\begin{cases}\sum_{m=1}^{|\mathcal{D}|}||\mathbf{S}^m-\mathbf{H}||_F^2, & m\in\mathcal{D}\\ \sum_{m=1}^{|\mathcal{E}|}||\mathbf{S}^m-\mathbf{G}||_F^2, & m\in\mathcal{E}\\ 0, & m\in\mathcal{U}\end{cases} \tag{3.3}$$

where $\mathcal{D}$, $\mathcal{E}$, and $\mathcal{U}$, respectively, represent the set of negative subjects, positive subjects, and unlabeled subjects. Moreover, $|\mathcal{D}|$ and $|\mathcal{E}|$, respectively, indicate the cardinality of $\mathcal{D}$ and $\mathcal{E}$.

Eq. (3.3) has at least two advantages: 1) preserving the global structure since all the subjects are close to their template and 2) outputting practical templates. However, Eq. (3.3) does not take the local structure of the data, which has been regarded as the complementary of the global structure of the data [121, 157]. In this chapter, we design $\mathcal{R}_2(\mathbf{S^1},...,\mathbf{S}^M)$ as follows

$$\mathcal{R}_2(\mathbf{S}^1,...,\mathbf{S}^M)=\frac{\sum_{m=1}^{M}\sum_{p\in\mathcal{G}(m)}||\mathbf{S}^m-\mathbf{S}^p||_F^2}{\sum_{m=1}^{M}\sum_{q\in\mathcal{F}(m)}||\mathbf{S}^m-\mathbf{S}^q||_F^2} \tag{3.4}$$

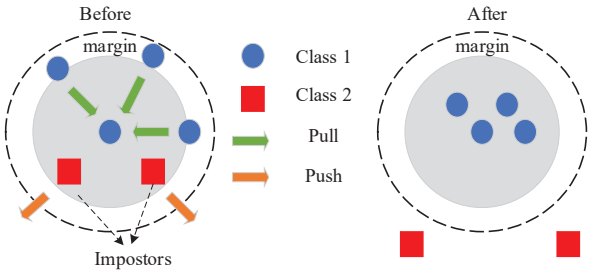


Figure 3.2: The visualization of Eq. (3.4). The left figure is the original neighborhood structure among one subject (*i.e.,* the centered point) and its neighbors. The right figure is the final status of the neighborhood structure about this subject after conducting the proposed multi-graph fusion method, where the subjects with the same label are close to each other and the subjects with different labels are far away from each other.

where $\mathcal{G}(i)$ and $\mathcal{F}(i)$, respectively, are the set of the near neighbors and the set of the distant neighbors, of the i-th subject. In the proposed framework, *i.e.,* semi-supervised learning, the training subjects include labeled subjects and unlabeled subjects, we denote the set $\mathcal{G}(i)$ of the i-th unlabeled subject as its k nearest neighbors including labeled subjects and unlabeled subjects, and the set $\mathcal{G}(i)$ of the i-th labeled subject as its k nearest neighbors with the same label to the i-th subject. We further define the set $\mathcal{F}(i)$ of the i-th unlabeled subject as its k furthest subjects including labeled subjects and unlabeled subjects, and the $\mathcal{F}(i)$ of the i-th labeled subject as its k nearest neighbors with different labels to the i-th subject. It is noteworthy that the value of k is insensitive in our experiments, so we fixed $k = 10$ for all subjects. Eq. (3.4) minimizes the ratio of two terms, similar to linear discriminative analysis [171, 194]. Specifically, the subjects have the same label with their nearest neighbors, while the subjects with far similarity have different labels. In this way, the local structure of the subjects is preserved. Figure 3.2 visualizes the process of Eq. (3.4). The optimization of Eq. (3.4) is very challenging, so we follow Theorem 1 in [148] to convert the minimization of Eq. (3.4) to minimize the following objective function:

$$\sum_{m=1}^{M}\Big(\sum_{p\in\mathcal{G}(m)}||\mathbf{S}^m-\mathbf{S}^p||_F^2-\lambda^m\sum_{q\in\mathcal{F}(m)}||\mathbf{S}^m-\mathbf{S}^q||_F^2\Big), \tag{3.5}$$

where λ^m can be updated as $\lambda^m=\frac{\sum_{p\in\mathcal{G}(m)}||\mathbf{S}^m-\mathbf{S}^p||_F^2}{\sum_{q\in\mathcal{F}(m)}||\mathbf{S}^m-\mathbf{S}^q||_F^2}$ in the implementation based on [148].

Compared to previous literature, Eq. (3.2) outputs the representation of every subject depending on other subjects as well as taking into account the following constraints, such as multi-graph information, and the preservation of the global and local structure among the subjects.

Theorem 1. *The global solution of the following general optimization problem*

$$\min_{\mathbf{v}\in\mathcal{C}}\frac{q(v)}{p(v)},\ \textit{where}\ q(v)>0\ (\forall v\in\mathcal{C}) \tag{3.6}$$

can be calculated by the root of the following function:

$$h(\lambda)=\min_{\mathbf{v}\in\mathcal{C}}q(v)-\lambda p(v), \tag{3.7}$$

given that $q(v) - \lambda p(v)$ is lower bounded.

Proof. Suppose v^* is the global solution of the problem in Eq. (3.18), and λ^* is the corresponding global minimal objective value, the formulation $\frac{q(v^*)}{p(v^*)} = \lambda^*$ holds, so $\forall v \in \mathcal{C}$, we have $\frac{q(v)}{p(v)} \geq \lambda^*$. By considering the characteristics of $p(v) > 0$, we can yield $q(v) - \lambda^* p(v) \geq 0$, which means:

$$\min_{v \in \mathcal{C}} q(v) - \lambda^* p(v) = 0 \quad \Longleftrightarrow \quad h(\lambda^*) = 0 \tag{3.8}$$

That is, the global minimal objective value λ^* in Eq. (3.18) is the root of the function $h(\lambda)$. Hence, the proof of Theorem 1 has been completed. □

3.2.2 Optimization

In this chapter, we employ the alternating optimization strategy [26] to optimize $\mathbf{S}^m$ $(m = 1, ..., M)$, $\mathbf{H}$, and $\mathbf{G}$.

(i) Update $\mathbf{S}^1, ..., \mathbf{S}^M$ by fixing $\mathbf{H}$ and $\mathbf{G}$

$\mathbf{S}^1, ..., \mathbf{S}^M$ include the representations of positive subjects, negative subjects, and unlabeled subjects, so we explain the optimization process one by one.

When m-th subject is a negative subject, we obtain the objective function with respect to $\mathbf{S}^m$ as follows:

$$\begin{aligned} &\min_{\mathbf{S}^m} \sum_{v=1}^{V} ||\mathbf{S}^m - \mathbf{A}^{m,v}||_F^2 + \alpha ||\mathbf{S}^m - \mathbf{H}||_F^2 + \\ &\beta(\sum_{p \in \mathcal{G}(m)} ||\mathbf{S}^m - \mathbf{S}^p||_F^2 - \lambda^m \sum_{q \in \mathcal{F}(m)} ||\mathbf{S}^m - \mathbf{S}^q||_F^2) \\ &s.t., \forall i, \mathbf{s}_{i,\cdot}^m \mathbf{1} = 1, s_{i,i}^m = 0, s_{i,j}^m \geq 0 \; if \; j \in \mathcal{N}(i), otherwise \; 0. \end{aligned} \tag{3.9}$$

where $\mathbf{S}^m$ represents the neighbor relationship between two brain regions (we represent $\mathbf{x}_i$ as the i-th brain region), $\mathbf{s}_i^m$ represents the relationship between $\mathbf{x}_i$ and other brain region $\mathbf{x}_j$ $(i \neq j)$. $\mathbf{s}_i^m$ is only related to $\mathbf{x}_i$ and $\mathbf{x}_j$ $(i \neq j)$, and is unrelated to $\mathbf{s}_j^m$. Therefore, in our optimization method, $\mathbf{s}_i^m$ is independent on $\mathbf{s}_j^m$ $(i \neq j)$, the objective function with respect to $\mathbf{s}_{i,\cdot}^m$ is:

$$\begin{aligned} \min_{\mathbf{s}_{i,\cdot}^m \mathbf{1}=1, s_{i,i}^m=0, s_{i,j}^m \geq 0} &\sum_{v=1}^{V} ||\mathbf{s}_{i,\cdot}^m - \mathbf{a}_{i,\cdot}^{m,v}||_2^2 + \alpha ||\mathbf{s}_{i,\cdot}^m - \mathbf{h}_{i,\cdot}||_2^2 + \\ &\beta(\sum_{p \in \mathcal{G}(m)} ||\mathbf{s}_{i,\cdot}^m - \mathbf{s}_{i,\cdot}^p||_2^2 - \lambda^m \sum_{q \in \mathcal{F}(m)} ||\mathbf{s}_{i,\cdot}^m - \mathbf{s}_{i,\cdot}^q||_2^2) \end{aligned} \tag{3.10}$$

Expanding Eq. (3.10), we have

$$\begin{aligned} \min_{\mathbf{s}_{i,\cdot}^m \mathbf{1}=1, s_{i,i}^m=0, s_{i,j}^m \geq 0} &\sum_{v=1}^{V} Tr(\mathbf{s}_{i,\cdot}^m - \mathbf{a}_{i,\cdot}^{m,v})^T (\mathbf{s}_{i,\cdot}^m - \mathbf{a}_{i,\cdot}^{m,v}) + \alpha Tr(\mathbf{s}_{i,\cdot}^m - \mathbf{h}_{i,\cdot})^T (\mathbf{s}_{i,\cdot}^m - \mathbf{h}_{i,\cdot}) + \\ &\beta(\sum_{p \in \mathcal{G}(m)} Tr(\mathbf{s}_{i,\cdot}^m - \mathbf{s}_{i,\cdot}^p)^T (\mathbf{s}_{i,\cdot}^m - \mathbf{s}_{i,\cdot}^p) - \lambda^m \sum_{q \in \mathcal{F}(m)} Tr(\mathbf{s}_{i,\cdot}^m - \mathbf{s}_{i,\cdot}^q)^T (\mathbf{s}_{i,\cdot}^m - \mathbf{s}_{i,\cdot}^q)) \end{aligned} \tag{3.11}$$

Expanding Eq. (3.11), we have

$$\begin{aligned}\min_{\mathbf{s}_{i,\cdot}^m\mathbf{1}=1,s_{i,i}^m=0,s_{i,j}^m\geq 0} &\sum_{v=1}^{V} Tr(\mathbf{s}_{i,\cdot}^{m^T}\mathbf{s}_{i,\cdot}^m - \mathbf{s}_{i,\cdot}^{m^T}\mathbf{a}_{i,\cdot}^{m,v} - \mathbf{a}_{i,\cdot}^{m,v^T}\mathbf{s}_{i,\cdot}^m + \mathbf{a}_{i,\cdot}^{m,v^T}\mathbf{a}_{i,\cdot}^{m,v}) \\ &+ \alpha Tr(\mathbf{s}_{i,\cdot}^{m^T}\mathbf{s}_{i,\cdot}^m - \mathbf{s}_{i,\cdot}^{m^T}\mathbf{h}_{i,\cdot} - \mathbf{h}_{i,\cdot}^T\mathbf{s}_{i,\cdot}^m + \mathbf{h}_{i,\cdot}^T\mathbf{h}_{i,\cdot}) \\ &+ \beta(\sum_{p\in\mathcal{G}(m)} Tr(\mathbf{s}_{i,\cdot}^{m^T}\mathbf{s}_{i,\cdot}^m - \mathbf{s}_{i,\cdot}^{m^T}\mathbf{s}_{i,\cdot}^p - \mathbf{s}_{i,\cdot}^{p^T}\mathbf{s}_{i,\cdot}^m + \mathbf{s}_{i,\cdot}^{p^T}\mathbf{s}_{i,\cdot}^p) \\ &- \lambda^m \sum_{q\in\mathcal{F}(m)} Tr(\mathbf{s}_{i,\cdot}^{m^T}\mathbf{s}_{i,\cdot}^m - \mathbf{s}_{i,\cdot}^{m^T}\mathbf{s}_{i,\cdot}^q - \mathbf{s}_{i,\cdot}^{q^T}\mathbf{s}_{i,\cdot}^m + \mathbf{s}_{i,\cdot}^{q^T}\mathbf{s}_{i,\cdot}^q))\end{aligned} \tag{3.12}$$

After that, we obtain:

$$\begin{aligned}\min_{\mathbf{s}_{i,\cdot}^m\mathbf{1}=1,s_{i,i}^m=0,s_{i,j}^m\geq 0} &Tr((V+\alpha+\beta(k-\lambda^m k))\mathbf{s}_{i,\cdot}^{m^T}\mathbf{s}_{i,\cdot}^m \\ &- 2(\sum_{v=1}^{V}\mathbf{a}_{i,\cdot}^{m,v^T} + \alpha\mathbf{h}_{i,\cdot}^T + \beta(\sum_{p=1}^{k}\mathbf{s}_{i,\cdot}^{p^T} - \lambda^m\sum_{q=1}^{k}\mathbf{s}_{i,\cdot}^{q^T})\mathbf{s}_{i,\cdot}^m \\ &+ \sum_{v=1}^{V}\mathbf{a}_{i,\cdot}^{m,v^T}\mathbf{a}_{i,\cdot}^{m,v} + \alpha\mathbf{h}_{i,\cdot}^T\mathbf{h}_{i,\cdot} + \beta(\sum_{p=1}^{k}\mathbf{s}_{i,\cdot}^{p^T}\mathbf{s}_{i,\cdot}^p - \lambda^m\sum_{q=1}^{k}\mathbf{s}_{i,\cdot}^{q^T}\mathbf{s}_{i,\cdot}^q))\end{aligned} \tag{3.13}$$

After conducting mathematical transformation, we have

$$\min_{\mathbf{s}_{i,\cdot}^m\mathbf{1}=1,s_{i,i}^m=0,s_{i,j}^m\geq 0} ||\mathbf{s}_{i,\cdot}^m - \mathbf{f}_{i,\cdot}^{m^-}||_2^2 \tag{3.14}$$

where

$$\mathbf{f}_{i,\cdot}^{m^-} = \frac{\sum_{v=1}^{V}\mathbf{a}_{i,\cdot}^{m,v^T} + \alpha\mathbf{h}_{i,\cdot}^T + \beta(\sum_{p=1}^{k}\mathbf{s}_{i,\cdot}^{p^T} - \lambda^m\sum_{q=1}^{k}\mathbf{s}_{i,\cdot}^{q^T})}{V+\alpha+\beta(k-\lambda^m k)} \in \mathbb{R}^{n\times 1} \tag{3.15}$$

We consider the Lagrangian Function of problem Eq. (3.14) as

$$\mathcal{L}(\mathbf{s}_{i,\cdot}^m, \varphi_1, \omega) = ||\mathbf{s}_{i,\cdot}^m - \mathbf{f}_{i,\cdot}^{m^-}||_2^2 - \varphi_1(\mathbf{s}_{i,\cdot}^m\mathbf{1} - 1) - \boldsymbol{\omega}\mathbf{s}_{i,\cdot}^m \tag{3.16}$$

where φ_1 is a Lagrange multiplier and $\boldsymbol{\omega}$ is a vector of nonnegative Lagrange multipliers. Conducting the differential with respect to $s_{i,j}^m$ and setting the results as zero, we obtain:

$$\frac{d\mathcal{L}}{ds_{i,j}^m} = \mathbf{s}_{i,j}^m - f_{i,j}^{m^-} - \frac{1}{2}\varphi_1 - \frac{1}{2}\omega_j = 0 \tag{3.17}$$

where $f_{i,j}^{m^-}$ is the j-th element of $\mathbf{f}_{i,\cdot}^{m^-}$. To facilitate the calculation, we set $\sigma_1 = \frac{1}{2}\varphi_1, \tau_j = \frac{1}{2}\omega_j$.

The complementary slackness of the Karush–Kuhn–Tucker (KKT) conditions [43] implies that the condition w$\tau_j = 0$ holds while $s_{i,j}^m$>0. Thus, we have the closed-form solution for $s_{i,j}^m$ as:

$$s_{i,j}^m = (f_{i,j}^{m^-} + \sigma_1)_+, j = 1, ..., n \tag{3.18}$$

where $f_{i,j}^{m^-}$ is the j-th element of $\mathbf{f}_{i,\cdot}^{m^-}$.

By following the same process from Eq. (3.9) to Eq. (3.18), we have

$$s_{i,j}^{m} = \begin{cases} (f_{i,j}^{m^-} + \sigma_1)_+, & m \in \mathcal{D} \\ (f_{i,j}^{m^+} + \sigma_2)_+, & m \in \mathcal{E} \\ (f_{i,j}^{m} + \sigma_3)_+, & m \in \mathcal{U} \end{cases} \tag{3.19}$$

where

$$\begin{cases} f_{i,j}^{m^+} = \frac{\sum_{v=1}^{V} \mathbf{a}_{i,j}^{m,v^T} + \alpha \mathbf{g}_{i,j}^T + \beta(\sum_{p=1}^{k} \mathbf{s}_{i,j}^{p^T} - \lambda^m \sum_{q=1}^{k} \mathbf{s}_{i,j}^{q^T})}{V + \alpha + \beta(k - \lambda^m k)} \\ f_{i,j}^{m} = \frac{\sum_{v=1}^{V} \mathbf{a}_{i,j}^{m,v^T} + \beta(\sum_{p=1}^{k} \mathbf{s}_{i,j}^{p^T} - \lambda^m \sum_{q=1}^{k} \mathbf{s}_{i,j}^{q^T})}{V + \beta(k - \lambda^m k)}. \end{cases} \tag{3.20}$$

σ_1, σ_2 and σ_3 are the Lagrange multipliers.

(ii)Update $\mathbf{H}$ and $\mathbf{G}$ by fixing $\mathbf{S}^1, ..., \mathbf{S}^M$

When $\mathbf{S}^1, ..., \mathbf{S}^M$ are fixed, the objective function with respect to $\mathbf{H}$ and $\mathbf{G}$ are:

$$\begin{cases} \min_{\mathbf{h}_{i,\cdot}\mathbf{1}=1, h_{i,i}=0, h_{i,j}\geq 0} \sum_{m=1}^{|\mathcal{D}|} ||\mathbf{S}^m - \mathbf{H}||_F^2 \\ \min_{\mathbf{g}_{i,\cdot}\mathbf{1}=1, g_{i,i}=0, g_{i,j}\geq 0} \sum_{m=1}^{|\mathcal{E}|} ||\mathbf{S}^m - \mathbf{G}||_F^2 \end{cases} \tag{3.21}$$

According to the KKT conditions, we have:

$$\begin{cases} h_{i,j} = (\hat{s}_{i,j}^{m^-} + \sigma_4)_+ \\ g_{i,j} = (\hat{s}_{i,j}^{m^+} + \sigma_5)_+ \end{cases} \tag{3.22}$$

where $\hat{s}_{i,j}^{m^-} = (\sum_{m\in\mathcal{D}} \mathbf{s}_{i,j}^{m^T})/|\mathcal{D}|$, $\hat{s}_{i,j}^{m^+} = (\sum_{m\in\mathcal{E}} \mathbf{s}_{i,j}^{m^T})/|\mathcal{E}|$, σ_4 and σ_5 are Lagrange multipliers.

The values of the Lagrange multipliers σ_1, σ_2, σ_3, σ_4, and σ_5, can be obtained by Lemma 1 [30]. For simplicity, we list the details of σ_3 as follows and the values of σ_1, σ_2, σ_4, and σ_5 can be obtained by similar principles.

Lemma 1. *By denoting $\mathbf{s}_{i,\cdot}^{m*}$ the optimal solution in Eq. (3.18), letting r and u be two indices, and $f_{i,r}^m > f_{i,u}^m$, if $s_{i,r}^{m*} = 0$, then $s_{i,u}^{m*}$ must be equal to zero.*

Based on Lemma 1, we can find some integers $\boldsymbol{I} = [\rho], 1 \leq \rho \leq n$ to meet the non-zero components of the sorted optimal solutions, *i.e.,*

$$\sigma_3 = \frac{1}{\rho}(\sum_{j}^{\rho} f_{i,j}^m - 1). \tag{3.23}$$

As a result, the optimal $\mathbf{s}_{i,\cdot}^{m*}$ can be described as $\mathbf{s}_{i,\cdot}^{m*} = max\{f_{i,j}^m - \rho, 0\}$, where the value of the optimal ρ is automatically obtained by Lemma 2 [30].

Algorithm 1: The pseudo of optimizing Eq. (3.2) .

Input: $\mathbf{B}^m$ $(m = 1, ..., M)$ and $\mathbf{y}$;
Parameters: C, α, and β;
Output: $\mathbf{S}^m$, $\mathbf{H}$, $\mathbf{G}$, and $\mathfrak{C}$;
Obtain $\mathbf{A}^{m,v}$ $(v = 1, ..., V)$ by $\mathbf{B}^m$;
Initialize $\mathbf{S}^m$ as the average of $\mathbf{A}^{m,v}$ $(v = 1, ..., V)$;
while $Eq.(3.2)$ *not converges* **do**
    Update $\lambda^m = \frac{\sum_{p \in \mathcal{G}(m)} ||\mathbf{S}^m - \mathbf{S}^p||_F^2}{\sum_{q \in \mathcal{F}(m)} ||\mathbf{S}^m - \mathbf{S}^q||_F^2}$;
    Update $\mathbf{H}$ and $\mathbf{G}$ via Eq. (3.22);
    Update $\mathbf{S}^m$ $(m = 1, ..., M)$ via Eq. (3.19);
end while
Obtain $\mathbf{X}$ by extracting the upper triangle of $\mathbf{S}^m$;
Run L1SVM on $\mathbf{X}$ and $\mathbf{y}$ to output the classifier $\mathfrak{C}$;

Lemma 2. *Let $\boldsymbol{\eta}$ represents the vector after sorting $\mathbf{f}_{i,\cdot}^m$ in a descending order, the number of strictly non-negative elements in $\mathbf{s}_{i,\cdot}^m$ is $\rho = max\{\eta_j - \frac{1}{j}(\sum_{i=1}^{j=n} \eta_i - 1) > 0\}$.*

Based on Lemma 2, the non-zero number in the *i*-th row $\mathbf{s}_{i,\cdot}^m$, *i.e.,* the number of brain regions connected to the *i*-th brain region, is different from the non-zero number in the *j*-th row $\mathbf{s}_{j,\cdot}^m$ $(i \neq j)$. It is noteworthy that previous sparse methods set the same number of brain regions connected to each brain region. Obviously, our method is more flexible, compared to previous methods.

3.2.3 Joint regions selection and disease diagnosis

Our proposed framework generates a sparse FCN $\mathbf{S}^m$ $(m = 1, ..., M)$ from two graphs, *i.e.,* a fully-connected FCN and a 1NN graph, for each subject. Moreover, we follow previous methods to transfer the matrix representation to its vector representation, *i.e.,* extracting the upper triangle part of the symmetric matrix $\mathbf{S}^m$ $(m = 1, ..., M)$ to form a row vector $\mathbf{x}_{m,\cdot} \in \mathbb{R}^{1 \times [n(n-1)/2]}$. As a result, we have the data matrix $\mathbf{X} \in \mathbb{R}^{M \times [n(n-1)/2]}$ and the corresponding label vector $\mathbf{y} \in \{-1, 1\}^{M \times 1}$.

Many existing studies separately conduct feature selection and disease diagnosis (*i.e.,* classification) [78]. The goal of feature selection is to remove the redundant features from high-dimensional data because the vector representation is a 4005-dimension vector for 90 nodes in our data sets. However, the optimal results of feature selection cannot guarantee to achieve the optimal classification in such two separate processes. In this paper, we employ L1SVM to jointly conduct feature selection and classification, where the result of feature selection will be iteratively updated by the optimized classifier so that finally outputting significant classification performance. We list the pseudo of our proposed functional connectivity analysis framework in Algorithm 1.

3.3 Experiments

We experimentally evaluated our proposed method, compared to six state-of-the-art methods, on three real neuro-disease data sets with the rs-fMRI data, in terms of binary classification.

3.3.1 Experimental setting

3.3.1.1 Data sets

The data set fronto-temporal dementia (FTD) contains 95 FTD subjects and 86 age-matched healthy control (HC) subjects. FTD was derived from the NIFD database managed by the frontotemporal lobar degeneration neuroimaging initiative. The data set obsessive-compulsive disorder (OCD) has 20 HC subjects and 62 OCD subjects. The data set Alzheimer's Disease Neuroimaging Initiative (ADNI) includes 59 Alzheimer's disease (AD) subjects and 48 HC subjects.

Imaging data acquisition. We used A 3.0-Tesla MR system (Philips Medical Systems, USA) equipped with an eight-channel phased-array head coil to collect all rs-fMRI data. The parameters of gradient-echo Echo-Planner Imaging (EPI) sequence were listed as follows: Field Of View (FOV) = $208 \times 180mm$, matrix = 104×90, 72 slices, TR = $720ms$, TE = $33.1ms$, Flip Angle (FA) = 52°, and multi-band factor = 8, 1200 frames, 15min/ru. The subjects' heads were fixed with a sponge pad for preventing head movement from affecting the experimental results. During the scanning, the subjects need to close eyes, keep relax, and stay awake.

Functional imaging data preprocessing. We used the DPARSF toolbox[2] to preprocess the rs-fMRI data. We first deleted the first 10 time points of each subject, and then conducted the steps including slice timing correction, motion correction, normalization and spatial smoothing, on the obtained rs-fMRI data, for adapting the subjects to the scanning environment.

For all imaging data, we followed the automated anatomical labeling (AAL) template [141] to construct the functional connectivity network for each subject with 90 nodes. The region-to-region correlation was measured by the Pearson correlation coefficient.

3.3.1.2 Comparison methods

The comparison methods include the baseline method L1SVM, three popular methods for neuro-disease diagnosis, *i.e.,* High-Order Functional Connectivity (HOFC) [179], Sparse Connectivity Pattern (SCP) [31], and Connectivity Network Analysis method with Discriminative Hub Detection (CNHD) [149], and two deep learning methods, *i.e.,* Simplify Graph Convolutional networks (SGC) [159], Deep Iterative and Adaptive Learning Graph Neural Networks (DIAL-GNN) [23]. We list the details of the comparison methods as follows.

- L1SVM extracts the upper triangle part of the FCN of each subject as its representation, and then employs the ℓ_1-norm regularization term to jointly conduct feature selection and classification.

[2]http://rfmri.org/DPARSF.

- HOFC learns the sparse FCN based on the fully-connected FCN, whose element is the Pearson correlation coefficient, by taking into account the high-order information of the subjects.
- SCP searches the sparse FCNs from the fully-connected FCNs to effectively explore the heterogeneity across the subjects by taking into account the inter-subject variability among the subjects.
- CNHD first constructs functional connectivity networks based on the rs-fMRI data, and then conducts feature extraction and the classification task in an unified framework.
- SGC first regards the upper triangle part of the fully-connected FCN of each subject as its representation as well as uses the fully-connected FCNs of all subjects to obtain the local structure of all subjects, and then designs a graph neural network by preserving the original local structure to update the representations of all subjects.
- DIAL-GNN first extracts the upper triangle part of the FCN of each subject as its representation to obtain a original graph, and then learns the new representation for each subject.

L1SVM, DIAL-GNN and SGC extract the upper triangle of the fully-connected FCN as the representation of each subject. The methods (*e.g.,* HOFC, SCP, CNHD, and our proposed method) designed different methods to transfer fully-connected FCNs to sparse FCNs, followed by extracting the upper triangle part of the sparse FCN as the representation of each subject. It is noteworthy that all methods can be directly applied for supervised learning but only three methods (*e.g.,* DIAL-GNN, SGC and our method) can be used for personalized classification.

3.3.1.3 Setting-up

In our experiments, we repeated the 10-fold cross-validation scheme 10 times for all methods to report the average results as the final results. In the model selection, we set $\alpha, \beta \in \{10^{-3}, 10^{-2}, ..., 10^{3}\}$ in Eq. (3.2), and fixed $k = 10$ since the value of k is insensitive to the result of Eq. (3.2). We further set $C \in \{2^{-10}, 2^{-9}, ..., 2^{10}\}$ for ℓ_1-SVM. We followed the related literature to set the parameters of the comparison methods so that they outputted the best results.

We designed the following experiments to evaluate all methods, *i.e.,* the classification performance of both supervised learning and personalized classification, the effectiveness of the proposed multi-graph fusion method, the effectiveness of feature selection and visualization of the selected brain regions, and the visualization of the templates produced by our method. The evaluation metrics include ACCuracy (ACC), SENsitivity (SEN), SPEcificity (SPE), and Area Under the ROC Curve (AUC). Besides, we conducted the paired-sample t-tests between our method and every comparison method, in terms of ACC, SEN, SPE, and AUC. Moreover, the symbols "*" and "**", respectively, indicate that our method has statistically significant difference with $p<0.05$ and $p<0.001$ on the paired-sample t-tests at 95% significance level, compared to the comparison method. We report the results in Table 3.1 - Table 3.3.

Table 3.1: Classification results (%) of all methods on FTD.

Methods	Accuracy	Sensitivity	Specificity	AUC
L1SVM	64.77 ± 3.89*	62.54± 1.56*	67.48 ± 2.87*	65.87± 1.95*
HOFC	79.37 ± 4.20*	78.59 ± 5.37*	82.16 ±4.29**	79.52 ± 3.88*
SCP	84.75 ± 4.20**	82.59 ± 3.77*	85.56 ± 3.87**	83.93 ± 5.43*
CNHD	83.59 ± 3.67**	81.29 ± 2.58*	85.67 ±3.33	82.67 ± 3.87*
DIAL-GNN	85.19 ± 1.59**	86.39 ± 1.14**	**85.92** ± **2.43**	84.33 ± 1.29**
SGC	84.55 ± 3.95**	84.86 ± 5.33*	84.59 ± 4.85	84.63 ± 4.33**
Proposed	**86.98** ± 3.06	**87.53** ± 3.89	84.14 ± 2.59	**87.93** ± 3.59

Table 3.2: Classification results (%) of all methods on OCD.

Methods	Accuracy	Sensitivity	Specificity	AUC
L1SVM	76.67 ± 3.26*	73.29 ± 5.24*	77.77 ± 2.69*	78.59 ± 1.88*
HOFC	83.92 ± 2.36*	83.24 ± 5.08**	84.11 ± 2.12*	83.17 ± 1.26**
SCP	85.83 ± 3.42**	85.52 ± 4.11*	86.80 ± 3.87*	86.93 ± 3.10*
CNHD	86.83 ± 4.05**	86.98 ± 3.87**	86.64 ±4.53**	84.56 ± 4.39*
DIAL-GNN	85.59 ± 1.55**	85.33 ± 2.58*	86.39 ±2.77**	85.93 ± 2.47*
SGC	87.06 ± 2.43	85.52± 5.26*	87.56 ± 4.55*	86.15 ± 5.01**
Proposed	**88.05** ± 4.21	**87.52** ± 4.15	**89.42** ± 3.56	**88.48** ± 4.33

Table 3.3: Classification results (%) of all methods on ADNI.

Methods	Accuracy	Sensitivity	Specificity	AUC
L1SVM	76.88 ± 4.25*	77.83 ± 3.58*	76.10 ± 2.85*	74.95 ± 2.36*
HOFC	80.25 ± 1.72*	78.89 ± 2.09*	81.35 ± 2.12**	81.26 ± 3.78**
SCP	84.89 ± 3.98**	85.14 ± 3.21*	84.80 ± 2.83*	84.89 ± 3.25**
CNHD	85.97 ± 4.36**	87.21 ± 5.21	84.70 ±3.95**	84.65 ± 4.47*
DIAL-GNN	86.97 ±1.76**	86.88 ± 2.77**	87.02 ± 1.33	87.68 ± 2.73*
SGC	86.96 ± 2.81**	88.24 ± 2.85	86.15 ± 3.66**	88.78 ± 4.69**
Proposed	**88.84** ± 3.22	**89.55** ± 1.85	**88.25** ± 2.49	**90.22** ± 3.4

3.3.2 Result analysis

3.3.2.1 Supervised learning

In the experiments of supervised learning, we used all labeled subjects as the training set. We report the results of all methods in Tables 3.1-3.3 and list our observations as follows.

First, our proposed method achieved the best classification performance on all three data sets, in terms of four evaluation metrics, followed by SGC, DIAL-GNN, CNHD, SCP, HOFC, and L1SVM. Moreover, our proposed method has statistically significant difference at 95% significance level, compared to most of comparison methods, in terms of evaluation metrics including ACC, SEN, SPE, and AUC. Specifically, our method on average improved by 2.17%, 1.71%, and 1.68%, respectively, compared to the best comparison method SGC, on FTD, OCD, and AD, for all evaluation metrics. The possible reasons are that (i) our multi-graph fusion method takes the inter-subject variability, the heterogeneity across subjects, and the discriminative ability into account to output homogenous and discriminative representation, and (ii) our proposed method jointly selects features (*i.e.,* brain regions) and conducts classification to avoid the influence of redundant features on high-dimensional data.

Second, L1SVM uses fully-connected FCNs, and other methods (*i.e.,* our proposed method, DIAL-GNN, SGC, SCP, and HOFC) learns sparse FCNs. As a result, L1SVM obtained the worst classification performance. For example, the worst method for learning sparse FCNs, *i.e.,* HOFC,

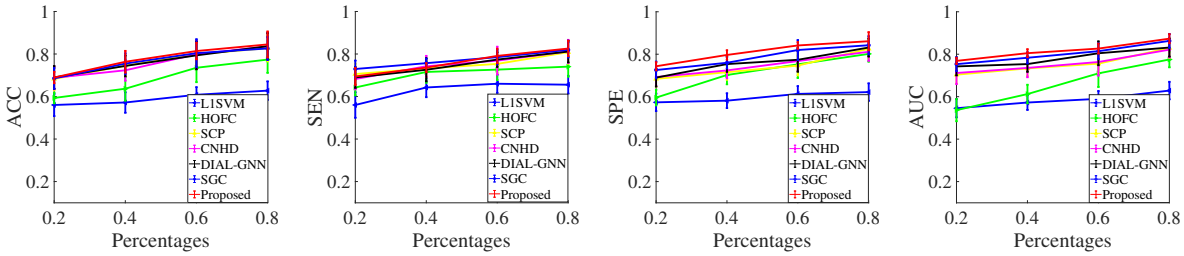


Figure 3.3: Classification results (mean ± standard deviation) of personalized classification on FTD.

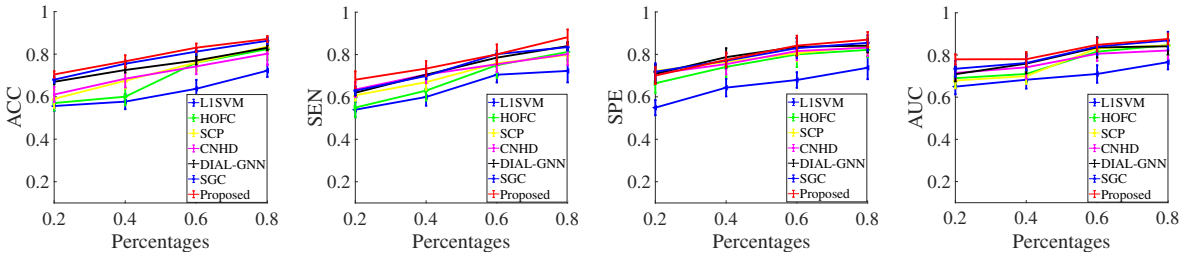


Figure 3.4: Classification results (mean ± standard deviation) of personalized classification on OCD.

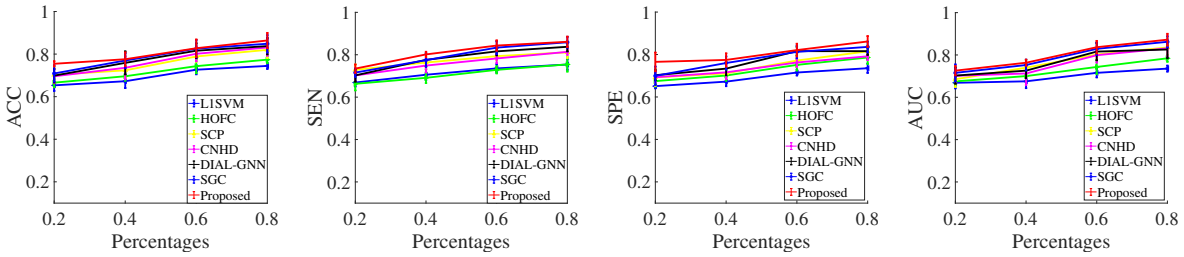


Figure 3.5: Classification results (mean ± standard deviation) of personalized classification on AD.

on average improved by 14.74%, 7.03%, and 3.99%, respectively, compared to L1SVM, on FTD, OCD, and AD, in terms of all four evaluation metrics. In particular, our proposed method fuses a fully-connected FCNs with a 1NN FCN from each subject to output a sparse FCN for every subject, followed by employing the L1SVM to conduct the classification task. On the contrary, L1SVM directly regards a fully-connected FCN as the representation for each subject to conduct the classification task. Moreover, our method on average improved by 16.98%, compared to L1SVM, in terms of Sensitivity, on all three data sets. This indicates the reasonability of sparse FCNs. That is, the sparse FCN is more suitable than the fully-connected FCN for conducting the FCN analysis on the rs-fMRI data.

Third, the methods (*e.g.,* HOFC, SCP, and our method) design different models to generate sparse FCNs. Specifically, they first generate the sparse FCNs in different ways and then convert the upper triangle parts of the derived FCNs as the new representation of the subjects. As a consequence, our method considers the heterogeneity across subjects to outperform other methods (*e.g.,* HOFC and SCP). This demonstrates that it is reasonable for taking into account the het-

erogeneity across subjects. In addition, CNHD, SGC, and DIAL-GNN also considers the heterogeneity across subjects and outperforms either HOFC or SCP. This verifies the importance of the consideration of the heterogeneity across subjects again. Furthermore, our proposed method is the only one to fuse the information from multiple FCNs so that achieving the best classification performance. This shows that our multi-graph fusion method is feasible because it can use the common and complementary information among multiple FCNs to output the discriminative representation of all subjects.

3.3.2.2 Personalized classification

To verify the effectiveness of our proposed semi-supervised method, we randomly selected different percentages of labeled subjects (*i.e.,* 20%, 40%, 60%, and 80%) from the whole data set as the training set. In this case, the methods (*i.e.,* L1SVM, HOFC, SCP, and CNHD) only used labeled subjects to train the classifiers, while the methods (*i.e.,* our method, SGC, and DIAL-GNN) used all subjects (*i.e.,* labeled subjects and unlabeled subjects) to train the classifiers. We report the classification results of all methods in Figures 3.3-3.5.

Similar to the scenarios of supervised learning, our proposed method still achieved the best performance, followed by SGC, DIAL-GNN, CNHD, HOFC, SCP and L1SVM, in terms of semi-supervised learning. Moreover, the paired-sample t-tests between our method and every comparison method showed that our proposed method has statistically significant difference at 95% significance level, compared to every comparison method, in terms of evaluation metrics including ACC, SEN, SPE, and AUC, at different label ratios on each data set. For example, our method on average improved by 1.97% and 14.91%, respectively, compared to the best comparison method SGC and the worst comparison method L1SVM, on three data sets, in terms of all evaluation metrics. Besides, we have other observations as follows.

By comparing the semi-supervised learning methods (*i.e.,* SGC, DIAL-GNN, and our method) with the supervised learning methods (*i.e.,* CNHD, HOFC, SCP, and L1SVM), the former methods outperformed the latter methods. Specifically, the former methods on average improved by 8.74%, 6.52%, 6.05%, and 8.34%, respectively, compared to the latter methods, on all three data sets with all different percentages of labelled subjects, in terms of ACC, SEN, SPE, AUC. This reason is that the semi-supervised learning methods use more data (*i.e.,* the unlabelled data) than the supervised learning methods during the training process. As a result, the semi-supervised learning methods may output more robust classifiers than the supervised learning methods. In particular, the improvement of the semi-supervised learning methods over the supervised learning methods achieves the maximum while the percentage of labeled subjects in the training set is small, *i.e.,* 20%. For example, the classification accuracy of our proposed method improved by on average 3.08%, 3.71%, 4.15%, and 3.29%, respectively, compared to the performance of the best method of supervised learning, *i.e.,* CNHD, in terms of 20%, 40%, 60%, and 80% of the percentage of labeled subjects in the training set, on all three data sets.

The percentage of labeled subjects in the training set is small, all methods achieved worse performance. Moreover, the more the percentage of labelled subjects is, the lower the improvement

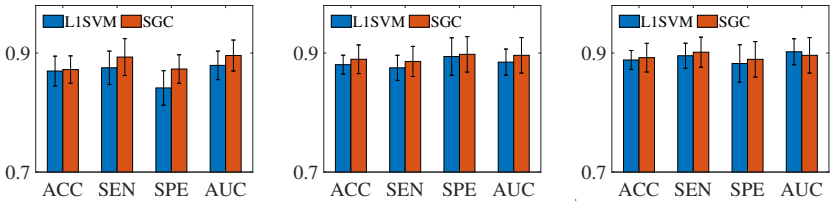
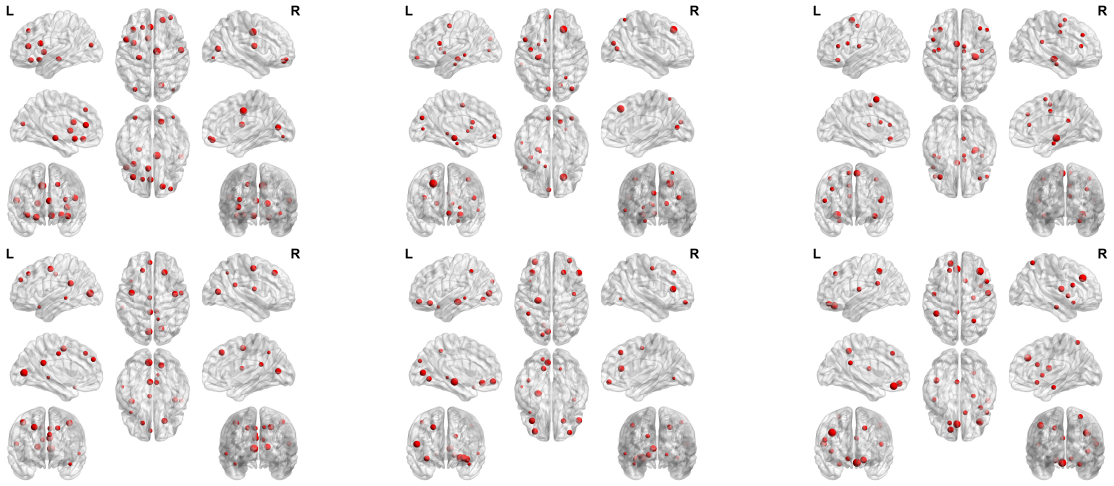


Figure 3.6: Classification results of L1SVM and SGC using the sparse FCNs produced by our method on FTD (left), OCD (middle), and ADNI (right).



(a) (a). FTD (b) (b). OCD (c) (c). ADNI

Figure 3.7: Visualization of top selected brain regions selected and the connected regions by L1SVM (upper) and our method (bottom) on FTD, OCD, and ADNI.

of our proposed method over the comparison methods is. For example, all methods achieved worse experiment results with only 20% label subjects for the training process. This contributes to the fact that it is difficult to build robust classifiers without sufficient labeled subjects. On the contrary, the classification performance increases with the increased percentage of the labeled subjects for the training process. For example, the classification accuracy of our method increased 7.41% from 20% to 40%, in terms of the percentage of labeled subjects, while improving by 4.47%, from 60% to 80%, on all three data sets. The main reason is that the limited labeled subjects is difficult to guarantee the discriminative ability of the classifiers.

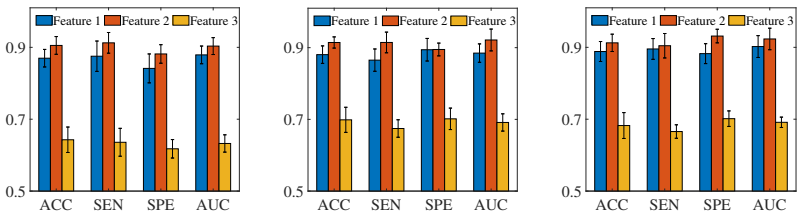
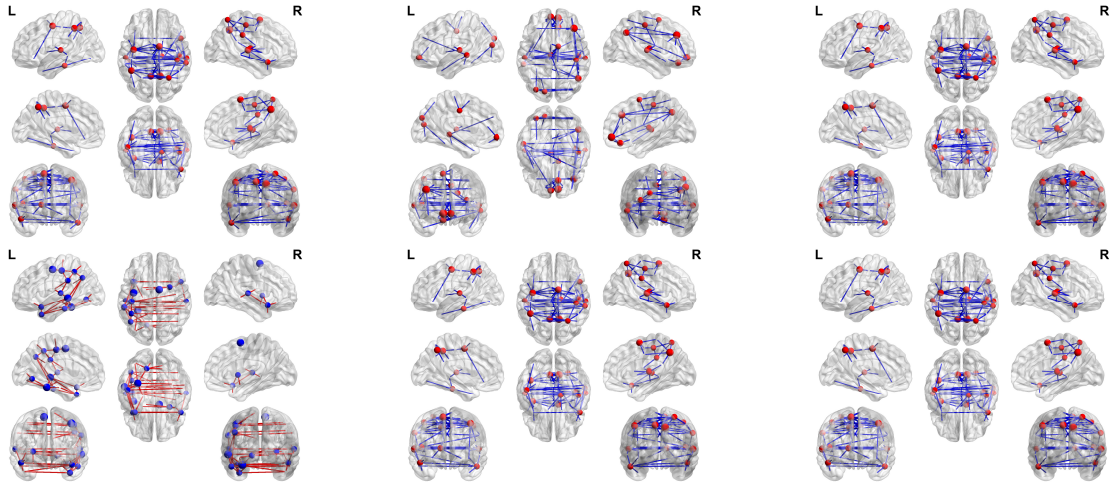


Figure 3.8: Classification results of L1SVM using three kinds of data (*i.e.,* Feature 1, Feature 2, and Feature 3) on FTD (left), OCD (middle), and ADNI (right).



(a) FTD (b) OCD (c) ADNI

Figure 3.9: Visualization of templates outputted by our method on FTD, OCD and AD (upper) and healthy control (bottom).

3.3.2.3 Multi-graph fusion effectiveness

The novelty of our multi-graph fusion method is to automatically learn both the common information and the complementary information among multiple FCNs. To verify the effectiveness of our proposed fusion method, we used our method to first generate a sparse FCN for each subject followed by extracting its vector representation (*i.e.,* the upper triangle part of the sparse FCN) as the new representation of the subjects, and then feed the new representation to the methods (*e.g.,* L1SVM and SGC) to output the classification performance. We report the experimental results in Figure 3.6. It is noteworthy that we only analyzed the best and the worst comparison methods due to the space limitations.

From Figure 3.6, the classification performance of the methods (*i.e.,* L1SVM and SGC) is better than the results of the corresponding methods in Tables 3.1-3.3. For example, L1SVM and SGC, respectively, on average improved by 16.2% and 4.3%, compared to the corresponding results in Tables 3.1-3.3, on all three data sets, in terms of all four evaluation metrics. This implies that the sparse FCNs outputted by our proposed multi-graph fusion method has strongly discriminative ability.

3.3.2.4 Feature selection effectiveness

In this section, we designed two kinds of experiments to investigate the effectiveness of the selected features by our method. Specifically, in our experiments, we repeated the 10-fold cross validation scheme 10 times, and thus outputting 100 subsets of the selected features. We further calculated the selected frequency of all features over all 4005 features, and then reported the features with the frequency over 90 out of 100 times as the top features. As a result, our method selected 1270, 898, 923 nodes out of 4005 nodes, respectively, on FTD, OCD, and AD, while L1SVM selected 1477, 1058, and 1213 nodes, respectively. It is noteworthy that each node or

feature is related to two brain regions.

We reported the top selected brain regions based on the selected features and plot the top selected brain regions of our method and L1SVM in Figure 3.7. Based on the visualization of the top selected brain regions, many selected regions from both our method and L1SVM have been verified related to the neuro-diseases. Specifically, in Figure 3.7(a), most of the nodes selected by our method occur in frontal and temporal lobes, which is consistent with the current neurobiological findings on FTD [27]. However, a large portion of nodes identified by L1SVM have low correlation with FTD. In Figure 3.7(b), our method finds the brain regions, such as orbital-frontal cortex, caudate, thalamus, which are included in the cortical-striato-thalamic circuits, and is considered as the theoretical neuroanatomical network of OCD [40, 41]. In particular, our method selected the brain regions throughout the whole brain because AD has been demonstrated to be associated with whole brain atrophy [117]. On the contrary, L1SVM only selected the frontal regions on the data set ADNI.

In our experiments, we first obtained three data sets based on the original data, *i.e.,* Feature 1, Feature 2, and Feature 3. Feature 1 represents the original data sets with high-dimensional data. Feature 2 is the data sets with the features selected by our method. Feature 3 is the data sets with the features unselected by our method. We fed these new data sets to L1SVM and reported the classification results in Figure 3.8. From Figure 3.8, Feature 2 achieved the best performance, followed by Feature 1 and Feature 3. Specifically, the classification accuracy of Feature 2 improved on average by 3.58%, compared to Feature 1, on all three data sets. The reason is that the original data contains redundant feature and noise, which affects classification performance. This illustrates (1) the effectiveness of the features selected by our method, and (2) feature selection can improve model performance.

3.3.2.5 Template visualization

In Eq. (3.3), we denoted $\mathbf{G}$ and $\mathbf{H}$, respectively, as the positive and negative template, to make the outputted representation containing discriminative ability. In this section, we visualized the templates in Figure 3.9 for all three data sets.

Obviously, the difference between the disease template and the healthy control template is significant. Moreover, the selected brain regions in the templates can be found as the top selection regions in Figure 3.7. This indicates the outputted templates can make our proposed method have discriminative ability as well as are possibly used for guiding either the parcellations for new subjects or measuring the group difference in the study of medical image analysis.

3.4 Discussion

In this section, we discuss time complexity of all methods, the variations of our proposed method with different k values, and the variations of our proposed with different initialization, the variations of our proposed with different numbers of graphs.

Table 3.4: Training time (Seconds) of all methods on three data sets.

Dataset	FTD	OCD	AD
L1SVM	1997s	951s	1248s
HOFC	3610s	1857s	2387s
SCP	2487s	1293	1639s
CNHD	2937	1433	2108s
DIAL-GNN	872s	407s	625s
SGC	668s	272s	438s
Proposed	2616s	1122s	1753s

Table 3.5: Classification results (ACC%) of our proposed method with different k on three data sets.

k	5	10	15	20	25
FTD	84.25 ± 3.89	86.98 ± 1.56	86.98 ± 2.87	86.10± 1.95	86.10± 1.95
OCD	87.53 ± 4.20	88.05 ± 5.37	88.05 ± 4.29	—	—
AD	87.71 ± 4.20	88.84 ± 3.77	87.53 ± 3.87	87.53 ± 5.43	88.06± 1.95

Table 3.6: Classification results (ACC%) of our proposed method on five different initializations

	Initialization 1	Initialization 2	Initialization 3	Initialization 4	our method
FTD	86.25± 1.58	85.87 ± 4.25	86.18 ± 3.11	85.23± 1.45	86.18± 1.95
OCD	88.05 ± 3.78	88.05 ± 2.56	88.05 ± 2.66	87.49 ± 2.77	88.58 ± 3.16
AD	88.18 ± 3.15	87.46 ± 2.97	89.38 ± 4.58	87.53 ± 5.43	88.18± 2.54

3.4.1 Time complexity analysis

The multi-graph generation is off-line. Hence, we ignore the calculation of the time complexity and the space complexity. The multi-graph fusion method takes a closed-form solution for the optimization of $\mathbf{S}^m$ $(m = 1, ..., M)$, $\mathbf{H}$ and $\mathbf{G}$. The time complexity of $\mathbf{S}^m$ is $O(Mn^2)$ and the time complexity of either $\mathbf{H}$ or $\mathbf{G}$ is $O(n^2)$, where M and n, respectively, represent the number of the subjects and the number of brain regions. Hence, the time complexity of our multi-graph fusion method is $O(lMn^2)$, *i.e.,* linear to the subject size, where l is the iteration number and is less than 50 in our experiments. Moreover, the proposed multi-graph fusion method needs to store $\mathbf{S}^m$ $(m = 1, ..., M)$, $\mathbf{H}$, and $\mathbf{G}$ in the memory with the space complexity $O(Mn^2)$. The time complexity of L1SVM is linear to the subject size, while its space time complexity is $O(Mn^2)$ [33]. Moreover, based on [33], L1SVM fast achieves convergence.

The time complexity of DIAL-GNN and SGC, respectively, are $\mathcal{O}(TdM^2)$ and $\mathcal{O}(M^2d)$, where M, T, $d = \frac{n(n-1)}{2}$, and n, are the number of samples, the iterations, the features, the brain regions, respectively. The time complexity of HOFC, SCP, CNHD, and L1SVM, respectively, are $\mathcal{O}(M(\omega-1)n^2+Md)$, $\mathcal{O}(Mn^2+Md)$, $\mathcal{O}(Mn^3)$ and $\mathcal{O}(Md)$, where ω is the window length.

More specifically, two deep learning methods (*i.e.,* DIAL-GNN and SGC) is quadratic to the sample size, while four traditional methods (*i.e.,* HOFC, SCP, CNHD, L1SVM, and our method) is linear to the sample size. However, in our data sets, the sample size is smaller than the number of the brain regions. Hence, two deep learning methods are faster than the traditional methods.

In Table 3.4, we report the training time of all methods on three data sets. HOFC needs the maximal training costs. Our proposed method needs the second most time as it includes the process of L1SVM. As a result, our proposed multi-graph fusion model is efficient, *i.e.,* linear to the sample size. For example, the time cost for our proposed multi-graph fusion model is 619 seconds, 171 seconds, and 505 seconds, respectively, for data sets FTD, OCD, and AD.

3.4.2 Sensitivity analysis of k values

We varied the values of as $k = \{5, 10, 15, 20, 25\}$ and reported the classification accuracy of our proposed method with different k values on three data sets in Table 3.5. It is noteworthy that the data set OCD only takes the values of k as $k = \{5, 10, 15\}$ as it only has 20 healthy control subjects. As a result, our proposed method is insensitive to the k values as the gap of two different scenarios is not significant in terms of classification accuracy. For example, the difference between the case '$k = 10$' and other cases (*i.e.,* $k \neq 10$) varied on average by 1.12%, 0.26%, 0.79%, on data sets FTD, OCD, and AD, respectively, in terms of classification accuracy.

3.4.3 Sensitivity analysis of initialization

In Algorithm 1, we initialize $\mathbf{S}^m$ $(m = 1, ..., M)$ as the average of $\mathbf{A}^{m,v}$ $(v = 1, ..., V)$, which makes the optimization of Eq. (3.2) converge within tens of iterations. Moreover, the result of Eq. (3.2) is insensitive to the initialization of $\mathbf{S}^m$ $(m = 1, ..., M)$.

In the experiment, we used the Matlab function rand(.) to generate two uniformly distributed random matrices (*i.e.,* Initialization 1, Initialization 2), and then used the Matlab function randn(.) to generate two random matrices that obey the Gaussian distribution (*i.e.,* Initialization 3, Initialization 4). In particular, our method set the average of $\mathbf{A}^{m,v}$ $(v = 1, ..., V)$ as the initialization of $\mathbf{S}^m$. From Table 3.6, our proposed method is insensitive to the initialization of $\mathbf{S}^m$. The main reason is that our optimization method is an iterative process that updates the value of $\mathbf{S}^m$ at each iteration, so that even a poor initialization can finally achieve reasonable results after many iterations. In addition, our initialization method converges faster than other initialization methods. The possible reason is our initialization $\mathbf{S}^m$ is the average of $\mathbf{A}^{m,v}$ $(v = 1, ..., V)$, which is closer to the final $\mathbf{S}^m$.

3.4.4 Effectiveness with different numbers of graphs

Our proposed method only explored the relationship between the fully-connected FCN and the extremely sparse FCN, *i.e.,* the 1NN FCN, aiming at learning the suitable connection number of the brain regions for each subject. To do this, our proposed method took into account the following issues such as noise, individual variability for each subject, and the inter-group heterogeneity across subjects. Actually, we can combine much sparse FCNs with the current two FCNs to learn the connection number. Thus, we added five more FCNs (*e.g.,* 3NN, 5NN, 8NN, 10NN, and 15NN) into the proposed objective function. As a result, the classification performance had no significant difference from the reported one in this work, *i.e.,* only on average improving by about 0.25%. The possible reason is that two FCNs (such as the fully-connected FCN and the 1NN FCN) are enough for our proposed method to automatically learn the connection number between 1 and n (where n is the node number).

3.5 Conclusion

In this chapter, we proposed a new framework for functional connectivity network analysis using the rs-fMRI data which can explore both the common and the complementary information among multiple FCNs for each subject to improve the discriminative ability of the learned representation from the rs-fMRI data. The experimental results on three real data sets verified the effectiveness of the proposed method, compared to four comparison methods, in terms of the classification performance.

This chapter has been published in the CORE rank A* journal, *i.e.,* Medical Image Analysis [37].

Chapter 4

Adaptive Multi-graph Learning Graph Convolutional Network

4.1 Introduction

Graph Convolutional Network (GCN) has been widely applied in the graph data such as social networks and citation networks as it conducts representation learning while taking the structure information of the data (*i.e.,* the graph information) into account [35, 64, 104, 199]. The GCN and its variants deal with the graph data by two key steps, *i.e.,* graph learning constructing a graph from the original data and representation learning improving the discriminative ability of the original representations [24]. The graph can be provided by the original data or constructed by the k nearest neighbors (kNN) method [163]. Current GCN methods are focusing on the representation learning, *i.e.,* designing different convolution operators to improve the effectiveness of representation learning whereas ignoring the graph learning. For example, Chen *et al.* proposed to conduct representation learning by multiple convolution operations [20] and Jiang *et al.* proposed graph convolution methods for feature propagation [63].

In real applications, due to the noise and outliers, the initial graph often has wrong connections, which can degrade the robustness of the subsequent representation learning [162,192,196]. Therefore, improving the quality of the initial graph can further improve the performance of the GCN model. The goal of graph learning is to output the high-quality graph for guaranteeing the quality of representation learning. The quality of graph learning can be affected by many factors, such as outliers and redundancy in the original data and the structure preservation.

First, the original data often contains noise and redundancy, which seriously affect the robustness of graph learning. In traditional machine learning methods, a number of solutions have been proposed to remove or reduce the influence of noise and redundancy, including feature selection and subspace learning. For example, Sun *et al.* employed the feature selection method to remove irrelevant and redundant features in classification tasks [126]. Salem *et al.* designed subspace learning techniques to remove the noise of the data [115]. However, current deep learning meth-

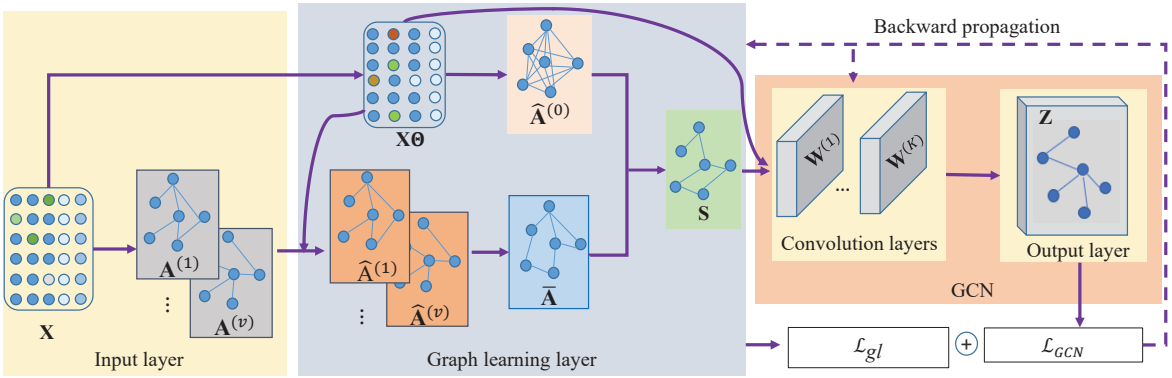


Figure 4.1: The flowchart of our proposed method. Specifically, our method uses the feature matrix $\mathbf{X}$ of the original data set to generate multiple sparse graphs $\mathbf{A}^{(v)}(v = 1, ..., V)$ as well as the low-dimensional data matrix $\mathbf{X\Theta}$, and then combines each $\mathbf{A}^{(v)}(v = 1, ..., V)$ with $\mathbf{X\Theta}$ to generate its local graph $\hat{\mathbf{A}}^{(v)}(v = 1, ..., V)$, followed by unifying all local graphs to generate the unified local graph $\bar{\mathbf{A}}$. The global graph $\hat{\mathbf{A}}^{(0)}$ learnt from the low-dimensional data matrix $\mathbf{X\Theta}$ is further integrated with $\bar{\mathbf{A}}$ to output the initial graph $\mathbf{S}$ for the GCN model. The learned graph $\mathbf{S}$ and the low-dimensional data $\mathbf{X\Theta}$ are then fed to a two-layer GCN model for representation learning. It is noteworthy that $\mathbf{S}$ keeps invariant in GCN (*i.e.,* the orange block) and varies in each epoch due to the back propagation process.

ods pay little attention to the issue of noise and redundancy on the original high-dimensional data, thus limiting the robustness of deep learning models.

Second, previous GCN methods focused on preserving the local structure of the data to construct the graph. However, the structural information of real-world data is complex, and a single characterization (*i.e.,* either the global structure or the local structure) may not be sufficient to capture the intrinsic structure of the data [37, 152, 175]. The studies of graph learning in traditional machine learning [37, 54] demonstrated that both the local structure preservation and the global structure preservation are important for graph learning as they provide the complementary information to each other [110, 196]. The local structure describes the internal organisation of the original data, while the global structure reflects the overall information of the original data. In graph representation learning, considering both the global and local structure of the data allows the graph representation model to be more robust to noisy and sparse data [91]. To our knowledge, few literature of previous GCN models focused on simultaneously considering the local structure preservation and the global structure preservation.

In this chapter, we propose a novel GCN method as shown in Figure. 4.1 to jointly conduct graph learning and representation learning in a unified framework by (1) fusing multiple local graphs on the low-dimensional space of the original high-dimensional data to obtain the unified local graph, where noise and redundancy are removed as much as possible, and (2) fusing the unified local graph and the global graph to output the initial graph for the GCN model. The first fusion step integrates the common information among multiple local graphs to guarantee the correctness of the edges in the unified local graph, *i.e.,* the correctness of the local structure

preservation. The second fusion step tries to find the missed edges in the local graphs from the global graph. Moreover, the proposed method simultaneously learns the projection matrix converting the original high-dimensional data to its low-dimensional space, two kinds of graph structures, and the representations of all samples. As a result, the update of each of them pushes the adaptive adjustment of representation learning, so that guaranteeing our proposed method to output discriminative representations.

Compared to previous methods, we list the main contributions of our method as follows.

First, we propose a new graph convolutional network method, which jointly conducts graph learning and representation learning from the low-dimensional space of the original data. To produce a high-quality graph, the proposed method explores the global structure and local structure of the data, with the aim of obtaining a correct connection for every node.

Second, we propose a novel graph fusion mechanism to integrate the multiple graphs information. Compare to traditional graph fusion method, the proposed graph fusion method (1) dynamically learns the weights for each graph during the fusion process so that the adversary effect of noise is effectively reduced and (2) conducts the process of graph fusion twice where the first fusion focuses on guaranteeing the correction of the edges and the second fusion focuses on finding the missed edges in the graphs from the original data.

4.2 Related work

4.2.1 Graph learning

Previous graph learning methods can be categorized into traditional graph methods and deep graph methods. Specifically, traditional graph methods usually use traditional machine learning methods to exploit simple structure information of the data, while deep graph methods employ deep learning models to capture complex structure among the samples. Moreover, each category includes two kinds of techniques, *i.e.,* static graph learning and dynamic graph learning, depending on the fact whether the graph structure is fixed or dynamically updated during the model training [175, 196].

Traditional graph methods include kNN graph, ϵ-graph, *etc.* For example, Takai *et al.* proposed using the hyper-graph structure to preserve the higher-order relations among samples, so that the samples in the same cluster are highly similar [129]. Since the graph in previous methods such as the kNN graph and the ϵ-graph is obtained from the original data as well as fixed in the process of the model training, so that it is easily influenced by noise and redundancy. To address the above issue, dynamic mechanism is widely embedded in graph learning methods to mine inherent correlation of the data in the low-dimensional feature space. For example, Xiong *et al.* proposed using the adaptive neighbor learning to capture the neighborhood information in the low-dimensional subspace [166], Li *et al.* explored the local structure information to adaptively assign each data point with optimal neighbors [87], Luo *et al.* incorporated the exploration of the local structure into the procedure of feature selection to learn the optimal graph [93]. Re-

cently, Wang *et al.* extended the conventional label diffusion to the label pair diffusion for image segmentation [150].

Deep graph methods take use of a multi-layer non-linear architecture to extract complex pattern from underlying data, aiming at learning complicated node representation or graph representation [104]. GCN is a well-known static deep graph method, where the graph is used to preserve the local structure of the data during the process of representation learning [75]. The graph quality is still the key issue in deep graph methods. Recently, Jiang *et al.* proposed integrating both the graph learning and the graph convolution in a unified network architecture so that the graph can be updated by the representation learning during the training process [62], Differently, Zhang *et al.* proposed using the self-attention mechanism to learn a parameterized adjacency matrix tailored to a specific task, and thus effectively improving the performance of the co-saliency detection task [180]. Peng *et al.* proposed to conduct a deep reverse graph learning based on the GCN for conducting graph learning on the low-dimensional space of original data [104].

4.2.2 Graph fusion

Graph fusion techniques are frequently used to generate a common graph structure by multiple feature representations or multiple graph structures. A typical example using fusion techniques is shown in [98] which enforces the features across all the views to construct a common similarity graph. Traditional graph fusion methods employed a graph learning method to perform graph fusion via minimizing the difference between the desired common graph and predefined multiple graphs [147, 195]. For example, Tang *et al.* proposed to capture both the common information and distinguishing knowledge across different views [132], Tong *et al.* conduct multi-modality classification using a nonlinear graph fusion method [139], Tang *et al.* take the traditional predefined graph matrices of different views as input, and learn an improved graph for each single view [131]. Recently, Lindenbaum *et al.* proposed using a random walk process to conduct multi-view dimensionality reduction [88]. In deep graph fusion methods, Zhuang *et al.* designed a dual GCN architecture to learn a robust node representation by simultaneously taking into account the local consistency and the global consistency [200], However, this method employed the fixed graph technique, and thus may limits the robustness of classification tasks.To perform consistent learning across multiple graphs, Jiang *et al.* proposed a fusion method by conducting dynamic GCN and consistent learning on multiple graphs in a unified framework [61].

4.3 Method

4.3.1 Motivation

Given an initial graph $\mathbf{A} \in \mathbb{R}^{n \times n}$ storing the graph structure of all samples of the feature matrix $\mathbf{X} \in \{\mathbf{x}_1, ..., \mathbf{x}_n\} \in \mathbb{R}^{n \times d}$ where n and d, respectively, indicate the number of the samples and the features, the GCN method taking both $\mathbf{A}$ and $\mathbf{X}$ as the inputs passes several hidden layers and one fully-connected layer to output the new representation $\mathbf{Z} \in \mathbb{R}^{n \times c}$ of $\mathbf{X}$ where c is the

number of the classes. Due to GCN have gained great popularity in tackling various analytics tasks on graph and network data

Specifically, the representation learned by the m-th layer of the GCN method can be obtained by:

$$\mathbf{H}^{(m)} = \sigma(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^{(m-1)}\mathbf{W}^{(m)}) \tag{4.1}$$

where $\mathbf{H}^{(m)} \in \mathbb{R}^{n\times d_m}$ denotes the outputted representation in the m-th hidden layer, $\mathbf{D} \in \mathbb{R}^{n\times n}$ is the diagonal matrix of $\mathbf{S}$ where d_{ii} is the summation of all elements in the *i*-th column of $\mathbf{A}$, $\mathbf{W}^{(m-1)} \in \mathbb{R}^{d_{m-1}\times d_m}$ is the weight matrix which needs to be trained in the (m)-th layer, d_m is the number of hidden units in the m-th layer, and $\sigma(.)$ is the activation function. The last layer of the GCN model is the classification layer with a softmax function:

$$\mathbf{Z} = softmax(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^{(M-1)}\mathbf{W}^{(M)}) \tag{4.2}$$

where $\mathbf{W}^{(M)} \in \mathbb{R}^{d_{M-1}\times c}$ denotes the weight matrix in the *M*-th hidden layer, M is the number of hidden layers. The weight parameters $(\mathbf{W}^{(1)}, ..., \mathbf{W}^{(M)})$ of the GCN are trained by minimizing the following cross-entropy loss function:

$$\mathcal{L}_{GCN} : -\sum_{i\in\mathcal{Y}}\sum_{j=1}^{c} y_{ij} ln z_{ij} \tag{4.3}$$

where $\mathcal{Y}$ is the label set, y_{ij} and z_{ij}, respectively, denote the ground truth and the prediction label for the j-th class and the i-th sample.

Recently, dynamic GCN methods have been proposed to jointly conducting graph learning and representation learning in a unified framework. As a result, the representation can be updated based on the optimized graph and thus guaranteeing to produce discriminative representation for the original data. For example, Jiang *et al.* employed the graph learning technique in traditional machine learning [100] to add one more regularization term (*i.e.,* $\mathcal{L}_{GL}$) [62] into the original GCN method, *i.e.,* $\mathcal{L}_{GCN}$ in Eq. (4.3). More specifically, given the initial graph $\mathbf{A} \in \mathbb{R}^{n\times n}$, the regularization term $\mathcal{L}_{GL}$ is defined as:

$$\begin{aligned}\mathcal{L}_{GL} : \min_{\mathbf{S}} &\sum_{i,j=1}^{n} \|\mathbf{x}_i - \mathbf{x}_j\|_F^2 s_{ij} + \lambda_1 \|\mathbf{A} - \mathbf{S}\|_F^2 \\ &s.t., \forall i, \mathbf{s}_i\mathbf{1} = 1, s_{ij} \geq 0.\end{aligned} \tag{4.4}$$

where $\mathbf{S} = \{s_{ij}\}_{i,j=1}^{n} \in \mathbb{R}^{n\times n}$ is the updated graph based on the initial graph $\mathbf{A}$ and $\|\cdot\|_F$ indicates the Frobenius norm. Finally, the objective function of the DGCN is:

$$\mathcal{L}_{DGCN} = \mathcal{L}_{GCN} + \lambda\mathcal{L}_{GL}. \tag{4.5}$$

where λ is the non-negative tuning parameter. Both graph $\mathbf{S}$ and the new representation $\mathbf{Z}$ are updated iteratively by minimizing Eq. (4.5). As a consequence, even though the original graph $\mathbf{A}$ is low-quality, DGCN can finally output discriminative representation of the original data $\mathbf{X}$.

The DGCN methods have been paying much attention on revising the regularization term $\mathcal{L}_{GL}$ to meet different requirements [23,38]. However, these methods have at least two drawbacks. First, both the initial graph $\mathbf{A}$ and the updated graph $\mathbf{S}$, *i.e.,* the second term in Eq. (4.5), are learnt from the original data. Second, many previous GCN methods only focused on exploring the local structure of the samples such as *k*NN graph and the ϵ-neighborhood graph, by ignoring the global structure of the samples. However, Silva *et al.* pointed out that both the local structure and the global structure are important for data analysis as they provide the complementary information to each other to improve the effectiveness of data analysis [167].

4.3.2 Multi-graph learning

In this section, we visualize the proposed framework in Figure 4.1 and introduce the proposed multi-graph fusion method in detail. First, we propose a projection matrix $\mathbf{\Theta} \in \mathbb{R}^{d \times d'}$ to convert the original data $\mathbf{X}$ to the low-dimensional data $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{\Theta}$ (where $d' < d$) where the noise and the redundancy are removed as much as possible. Second, we explore the local and global structure of the data through our proposed multi-graph fusion method. Specifically, we use multiple *k*NN graphs $\hat{\mathbf{A}}^{(v)}(v = 1, ..., V)$ (where V is the number of graphs) to learn the local structure of the data, and employ the self-representation method to learn the global graph $\mathbf{A}^{(0)}$ by preserving the global structure of all samples. Compared with previous graph learning methods, our multi-graph fusion method can capture the diversity of the graph structure in $\mathbf{X}$, because the multi-graph structure can provide rich edge information. Finally, we integrate the graph learning process and the representation learning process into a framework to realize the dynamic update of the graph structure and the data representation.

4.3.2.1 Initial graph generation

We employ the *k*NN graph method to generate multiple initial graphs. Specifically, the similarity between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is first defined as:

$$a_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma}} \tag{4.6}$$

where σ is a non-negative tuning parameter. After calculating the similarity of all samples, we then keep the similarities of the top-*k* neighbors for each sample and set other similarities as zeros, to finally obtain a sparse *k*NN graph $\mathbf{A}$, *i.e.,* an initial graph for $\mathbf{X}$.

Obviously, we obtain V initial graphs $\mathbf{A}^{(v)}$ $(v = 1, ..., V)$ by setting different k values. The graphs $\mathbf{A}^{(v)}$ $(v = 1, ..., V)$ preserve the local structure as each node in the graph connects only k nearest neighbors. We let $\mathbf{A}^{(v)} = (\mathbf{A}^{(v)} + \mathbf{A}^{(v)^T})/2$ to guarantee that the learned graphs are symmetric.

4.3.2.2 Local structure learning

The local structure of the data, *i.e.,* the local neighborhood relationship of a data set, is important to maintain the manifold structure of high-dimensional data [18] and is often characterized

Algorithm 2: The pseudo of our proposed method.

Input: Feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, the label $\mathcal{Y}$, λ_1, λ_2, λ_3, η, and β;
Output: The model weights and classifying unlabelled samples;
Generate multiple graphs $\mathbf{A}^{(v)} \in \mathbb{R}^{n \times n}$ (v=1,...,V) by the *k*NN graph method from $\mathbf{X}$;
Initialize the model weights $(\mathbf{\Theta}, \mathbf{W}^{(1)}, ..., \mathbf{W}^{(M)})$;
while $epoch < 5000$ **do**
  $epoch = epoch + 1$;
  $\hat{\mathbf{A}}^{(v)}(v = 1, ..., V) \leftarrow \{\mathbf{X}, \mathbf{\Theta}, \mathbf{A}^{(v)}\}$ by Eq. (4.7);
  $\hat{\mathbf{A}}^{(0)} \leftarrow \{\mathbf{X}, \mathbf{\Theta}\}$ by Eq. (4.8);
  $\mathbf{S} \leftarrow \{\hat{\mathbf{A}}^{(v')}(v = 0, 1, ..., V)\}$ by Eq. (4.11);
  $\mathbf{H}^{(m)}(m = 1, ..., M) \leftarrow \{\mathbf{S}, \mathbf{H}^{(m-1)}, \mathbf{W}^{(m-1)}\}$ by Eq. (4.1);
  $\mathbf{Z} \leftarrow \{\mathbf{S}, \mathbf{W}^{M}, \mathbf{H}^{M}\}$ by Eq. (4.2);
  $\mathcal{L} \leftarrow \{\mathbf{S}, \hat{\mathbf{A}}^{v'}(v' = 0, 1, ..., V), \mathbf{Z}, \mathcal{Y}\}$ by Eq. (4.13);
  Back-propagate $\mathcal{L}$ to update the model weights $(\mathbf{p}, \mathbf{\Theta}, \mathbf{W}^{(1)}, ..., \mathbf{W}^{(M)})$;
end while

through the *k* nearest neighbors of each data point in this work. Traditional local structure learning methods include Locality Preserving Projection (LPP) and Locally Linear Embedding (LLE), *etc.* However these methods have the disadvantages as follows. First, it is difficult to determine the value of *k*. Second, the initial *k*NN graphs are learnt from the original data which contains noise and redundancy. To overcome these problems, we propose to construct graphs in a low-dimensional feature space and integrates the optimization of graph structure and learning tasks in a unified framework, and thus can learn an optimal graph representation for GCN learning. To this end, we propose the following objective function to conduct the local structure learning:

$$\begin{aligned} \min_{\hat{\mathbf{A}}^{(v)}, \mathbf{\Theta}} & \sum_{i,j=1}^{n} \|\mathbf{x}_i\mathbf{\Theta} - \mathbf{x}_j\mathbf{\Theta}\|_F^2 \hat{a}_{ij}^{(v)} + \lambda_1 \|\hat{\mathbf{A}}^{(v)} - \mathbf{A}^{(v)}\|_F^2 \\ & s.t., \forall i, \hat{\mathbf{a}}_i^{(v)}\mathbf{1} = 1, \hat{a}_{ij}^{(v)} \geq 0. \end{aligned} \tag{4.7}$$

where λ_1 is a non-negative tuning parameter and $\mathbf{1}$ indicate the all-one-element vector. $\hat{\mathbf{A}}^{(v)} \in \mathbb{R}^{n \times n}$ is the updated graph matrix of the initial graph $\mathbf{A}^{(v)}$ and $\mathbf{\Theta} \in \mathbb{R}^{d \times d'}$ is the projection matrix which converts the high-dimensional data $\mathbf{X}$ to the low-dimensional space. In Eq. (4.7), the graph $\hat{\mathbf{A}}^{(v)}$ and the projection matrix $\mathbf{\Theta}$ are iteratively updated. Regarding that all the initial graphs are learnt from the original data which may contain noise and redundancy, we use the low-dimensional data $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{\Theta}$ to update them.

4.3.2.3 Global structure learning

In Eq. (4.7), $\hat{\mathbf{A}}^{(v)}$ only takes the local structure of the data into account. Recent literature shows that the preservation of the global structure of the data is of great importance for some machine learning tasks, *i.e.,* feature selection and classification. The reason is that the global

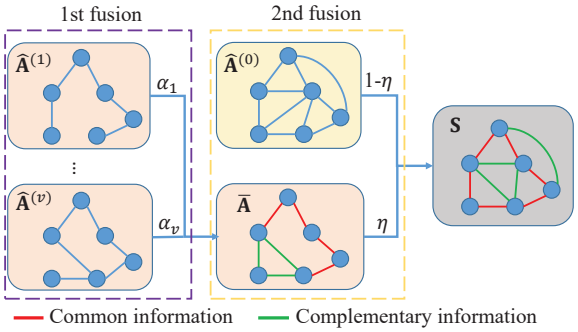


Figure 4.2: Visualization of the proposed multi-graph fusion method. Specifically, the first fusion (*i.e.,* the purple dot rectangle) outputs the common and complementary information among the local graphs but may have missed edges. The second fusion (*i.e.,* the yellow dot rectangle) outputs the common and complementary information among the local graphs and the global graph as well as adds the missed edges in the first fusion.

structure effectively contains the discriminative information, which can provide the complementary information different from the local structure preservation to capture the intrinsic structure of the data [136]. Self-representation property has been applied to many real applications and by demonstrating its capability in capturing the global structure of data [130]. Specifically, the self-representation property assumes that each data point can be linearly reconstructed from weighted combinations of all other data points, *i.e.,* $\mathbf{x}_i = \hat{\mathbf{a}}_i^{(0)}\mathbf{X} + \mathbf{e}$, where $\hat{\mathbf{a}}_i^{(0)}$ indicates a weight coefficient vector between $\mathbf{x}_i$ and $\mathbf{X}$, and the vector $\mathbf{e}$ is the noise bias. To this end, we propose the following objective function to conduct the global structure learning:

$$\begin{aligned} \min_{\hat{\mathbf{A}}^{(0)},\mathbf{\Theta}} \quad & \|\mathbf{X\Theta} - \hat{\mathbf{A}}^{(0)}\mathbf{X\Theta}\|_F^2 + \lambda_2\|\hat{\mathbf{A}}^{(0)}\|_1 \\ & s.t., \forall i, \hat{\mathbf{a}}_i^{(0)}\mathbf{1} = 1, \hat{a}_{ij}^{(0)} \geq 0. \end{aligned} \tag{4.8}$$

where λ_2 is a non-negative tuning parameter. In Eq. (4.8), the first term is used to generate a dense graph $\hat{\mathbf{A}}^{(0)}$, but the ℓ_1-norm on the global graph $\hat{\mathbf{A}}^{(0)}$ pushes to generate sparse representation while preserving the global structure.

4.3.2.4 Local and global preservation learning

Eq. (4.7) and Eq. (4.8), respectively, conduct the local structure learning and the global structure learning. Moreover, two kinds of learning are based on the low-dimensional data $\mathbf{X\Theta}$. Hence, in this paper, we integrate the local structure learning and the global structure learning with the projection matrix learning to have:

$$\begin{aligned} \mathcal{L}_{gl}: \min_{\hat{\mathbf{A}}^{(0)},\hat{\mathbf{A}}^{(1)},...,\hat{\mathbf{A}}^{(V)},\mathbf{\Theta}} & \sum_{v=1}^{V} tr(\mathbf{\Theta}^T\mathbf{X}^T\mathbf{L}^{(v)}\mathbf{X\Theta}) \\ & + \lambda_1\|\mathbf{X\Theta} - \hat{\mathbf{A}}^{(0)}\mathbf{X\Theta}\|_F^2 \\ & + \lambda_2\sum_{v=1}^{V}\|\hat{\mathbf{A}}^{(v)} - \mathbf{A}^{(v)}\|_F^2 + \lambda_3\|\hat{\mathbf{A}}^{(0)}\|_1 \\ & s.t., \forall i, \hat{\mathbf{a}}_i^{(v')}\mathbf{1} = 1, \hat{a}_{ij}^{(v')} \geq 0, (v' = 0, 1, ..., V). \end{aligned} \tag{4.9}$$

where $\mathbf{L}^{(v)} = \mathbf{D}^{(v)} - \hat{\mathbf{A}}^{(v)}$ is the Laplacian matrix of $\hat{\mathbf{A}}^{(v)}$. $\mathbf{1}$ is a vector whose elements are one and $tr(\cdot)$ is the trace operator. There are $(V+2)$ variables in Eq. (9), *i.e.,* $\hat{\mathbf{A}}^{(v')}(v' = 0, ..., V)$ and $\mathbf{\Theta}$, where $\mathbf{\Theta}$ is the trainable parameter update by the back-propagate process, $\hat{\mathbf{A}}^{(v')}(v' = 0, ..., V)$ are updated by Eq. (10). The details of the optimization of Eq. (9) are list in Algorithm 2.

First, if we fix $\mathbf{\Theta}$, the optimization with respect to $\hat{\mathbf{A}}^{(v')}(v = 1, ..., V)$ is Eq. (4.7). To reduce the complexity, we follow [62] to approximately optimize $\mathbf{p} \in \mathbb{R}^{r\times 1}$ by

$$\hat{a}_{ij}^{(v)} = \frac{exp(ReLU(\mathbf{p}^T|\mathbf{x}_i\mathbf{\Theta} - \mathbf{x}_j\mathbf{\Theta}|))}{\sum_{j=1}^{n} exp(ReLU(\mathbf{p}^T|\mathbf{x}_i\mathbf{\Theta} - \mathbf{x}_j\mathbf{\Theta}|))} \tag{4.10}$$

where $ReLU(.)$ is the activation function. If ${a_{ij}}^{(v)}$ is unavailable, we can set ${a_{ij}}^{(v)} = 1$ in the above update rule. After optimizing Eq. (4.9), we obtain the updated graphs $\hat{\mathbf{A}}^{(v')}$ $(v' = 0, 1, ..., V)$ on the low-dimensional features $\mathbf{X\Theta}$, rather than on the high-dimensional original feature matrix as in previous DGCN methods. Moreover, the projection matrix and the graphs are iteratively optimized, and thus Eq. (4.9) takes into account the local structure and the global structure to conduct dynamic graph learning.

4.3.3 Objective function

With the assumption that both the local graph structure and the global graph structure can provide complementary information different from others, we combine the learned local graphs and the global graph to obtain the input graph $\mathbf{S}$ for the GCN method as follows:

$$\mathbf{S} = \eta\bar{\mathbf{A}} + (1 - \eta)\hat{\mathbf{A}}^{(0)} \tag{4.11}$$

where $\bar{\mathbf{A}} = \sum_{v=1}^{V} \alpha_v\hat{\mathbf{A}}^{(v)}$, α_v denotes the weight (importance) of $\hat{\mathbf{A}}^{(v)}$ and η is a tuning parameter, which can be used to control the weight between two different kinds of graphs.

In Eq. (4.11), we combine multiple local graphs with a global graph by two weights, *i.e.,* α and η. The parameter η can be tuned in the implementation and the value of the parameter α_v can be obtained by

$$\alpha_v = \frac{exp(\sum_{ij} \hat{a}_{ij}^{(v)}\hat{a}_{ij}^{(0)})}{\sum_{v=1}^{V} exp(\sum_{ij} \hat{a}_{ij}^{(v)}\hat{a}_{ij}^{(0)})} \tag{4.12}$$

In Eq. (4.12), each local graph has individual distributions. Moreover, the larger the value of α_v, the more important the $\hat{\mathbf{A}}^{(v)}$ is.

Finally, we list our proposed objective function as follows:

$$\mathcal{L} = \mathcal{L}_{GCN} + \beta\mathcal{L}_{gl} \tag{4.13}$$

where β is a non-negative tuning parameter.

The proposed method in Eq. (4.13) conducts graph fusion twice, as shown in Fig. 4.2. The first fusion combines all sparse graphs to preserve the local structure and the parameter $\boldsymbol{\alpha}$ automatically learns the weight of every graph. Specifically, if some edges are only found in a small part

of all $\hat{\mathbf{A}}^{(v)}(v = 1, ..., V)$, the first fusion may automatically assign them with a small or even zero weight. Hence, they can be regarded as noisy edges and are removed out of $\bar{\mathbf{A}}$. If some edges are found in some sparse graphs with large weights, the first fusion may keep them in $\bar{\mathbf{A}}$ as the complementary information across the local graphs. If some edges are found in most graphs, the first fusion regards them as the common information among all local graphs kept in $\bar{\mathbf{A}}$. As a result, $\bar{\mathbf{A}}$ is a high-quality graph preserving the local structure, but could still miss some edges. The second fusion integrates the local graph $\bar{\mathbf{A}}$ and the global graph $\hat{\mathbf{A}}^{(0)}$ to address this issue as follows. Specifically, the common information (*i.e.,* the common edges between $\bar{\mathbf{A}}$ and $\hat{\mathbf{A}}^{(0)}$) and the complementary information (*i.e.,* the edges appearing in either $\bar{\mathbf{A}}$ or $\hat{\mathbf{A}}^{(0)}$) are outputted. Hence, our proposed method conducts the graph fusion twice to guarantee the quality of the learned graph $\mathbf{S}$.

4.3.4 Time complexity

The proposed model has $V + 1$ trainable graphs *i.e.,* $\hat{\mathbf{A}}^{(v')}(v' = 0, ..., V)$, the time complexity of learning an graph matrix is $\mathcal{O}(n^2 d')$, where n is the number of samples, d' is the number of features of the original data after dimensionality reduction. However, $V+1$ graph matrices can be trained at the same time in our proposed method, thus the total time complexity of graph learning in our proposed method is $\mathcal{O}(n^2 d')$. The time complexity of GCN is $\mathcal{O}(Mn^2 d' + Mnd'^2)$, where M is the number of hidden layers, hence the overall time complexity of our proposed model is $\mathcal{O}(n^2 d' + Mn^2 d' + Mnd'^2)$.

4.4 Experimental analysis

We conducted extensive experiments on twelve real data sets to compare our method with seven comparison methods in terms of semi-supervised node classification.

4.4.1 Experiment setting

4.4.1.1 Data sets

In our experiments, the used public data sets included four network data sets (*i.e.,* Citeseer, Cora, Pubmed and Wiki-CS), seven image data sets (*i.e.,* CIFAR10, SVHN, MNIST, Scene15, AWA, Handwritten and Caltech) and a text data set (*i.e.,* BBC). The details of all used data sets are listed as follows.

- **Citeseer** includes 3327 nodes distributed in 6 classes, where each node is represented by a 3703-dimensional feature descriptor.
- **Cora** has 2708 nodes which represented by a 1433-dimensional features. Moreover, all samples were classified into 6 classes.
- **Pubmed** includes 19717 nodes and each node has 500 features. Moreover, all nodes were classified into 3 classes.

- **Wiki-CS** has 11701 nodes distributed in 10 classes, where each node is represented by a 300-dimensional features.
- **CIFAR10** includes 50000 natural images distributed in 10 different classes.
- **SVHN** has 5000 images, where each class has 500 images.
- **MNIST** consists of 5000 images of hand-written digits from '0' to '9' in our experiments. Moreover, each class has 500 images.
- **Scene15** includes 4485 image distributed in 15 different classes. We followed [65] to obtain the feature of the samples.
- **BBC** consists of 2225 news documents within 5 categories, *i.e.,* business, entertainment, politics, sport and technique.
- **Handwritten** extracts normalized bitmaps of handwritten digits from a preprinted form. In our experiments, it has 2000 images which are in 10 classes.
- **Caltech** is an image data set including 1474 images which are distributed in 7 classes.
- **AWA** describes 50 animals by 4000 images. Each image is represented by 2688 dimensional color histogram features.

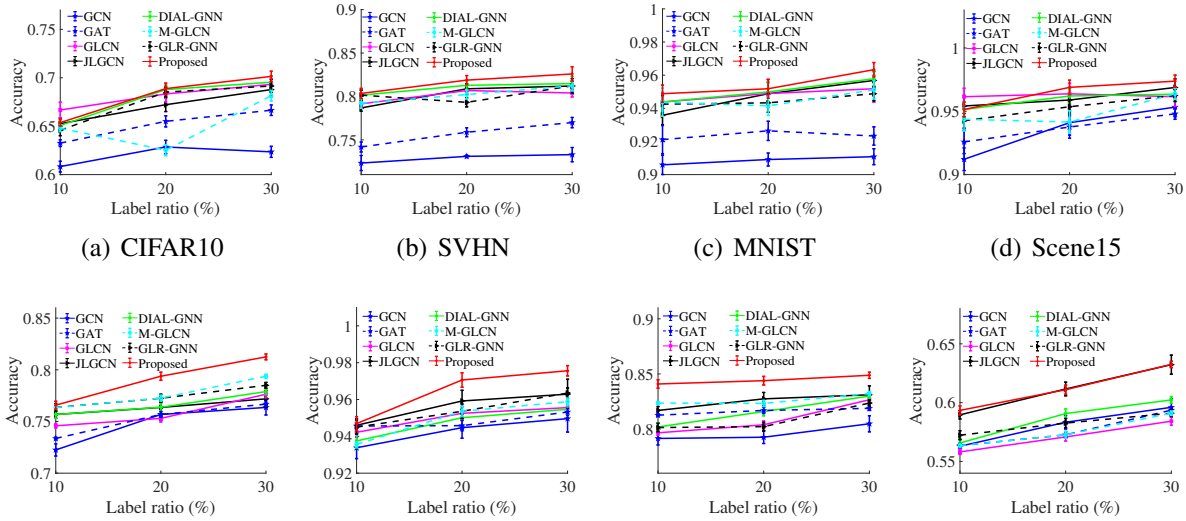
4.4.1.2 Comparison methods

The comparison methods included one baseline method, *i.e.,* GCN [75], and six state-of-the-art method, *i.e.,* Graph Attention Network (GAT) [142], Joint Learning Graph Convolutional Network (JLGCN) [133], Graph Learning Convolution Networks (GLCN) [62], Deep Iterative and Adaptive Learning Graph Neural Network (DIAL-GNN) [23]), Multiple Graph Learning, Convolutional Networks (M-GLCN) [61] and Graph Structure Learning for Robust Graph Neural Networks (GLR-GNN) [68].

- **GCN** encodes the feature information and the structure information of the data by the defined graph convolution operators. Moreover, it includes two convolutional layers (each of which had 20 units), followed by the softmax normalization for the label prediction.
- **GAT** uses mask self-attention layers to assign different weights to different nodes within a neighborhood without relying on prior knowledge of the graph structure.
- **JLGCN** jointly learns the graph structure and feature representation in the GCN framework. Specifically, it optimizes an underlying graph kernel via distance metric learning with the Mahalanobis distance. Moreover, the metric matrix is decomposed into a low-dimensional matrix and a graph Laplacian regularizer.
- **GLCN** integrates graph learning operations and traditional graph convolution structures in a unified network. To do this, it designs a graph learning regularization term to learn the optimal graph structure.

- **DIAL-GNN** casts the graph structure learning problem as a similarity metric learning problem as well as leverages an adapted graph regularization for controlling smoothness, connectivity and sparsity of the generated graph. DIAL-GNN designs a novel iterative method for searching for a hidden graph structure that augments the initial graph structure.
- **M-GLCN** jointly learns the multi-graph structure and the feature representation in the GCN framework. To do this, it designs a multi-graph learning mechanism to learn the internal relationship among the multi-graph structure.
- **GLR-GNN** targets on exploring the graph properties of sparsity, low rank and feature smoothness, aiming at designing robust graph neural networks. To do this, it uses a regularization term to learn the clean graph structure from the noisy graph data and jointly learn parameters for the robust graph neural network and the clean structure.

GCN conducts representation learning by preserving the local structure in the initial graph, which is invariant in the whole training process. The methods (*e.g.,* JLGCN, GLCN, GLR-GNN, M-GLCN and DIAL-GNN) jointly conduct graph learning and representation learning by designing different methods to improve the quality of the initial graph. GAT uses attention coefficients to combine its features with the features from its neighborhood to obtain a new representation.



(a) CIFAR10 (b) SVHN (c) MNIST (d) Scene15

(e) BBC (f) Handwritten (g) Caltech (h) AWA

Figure 4.3: Classification accuracy of all methods on eight data sets.

4.4.1.3 Setting-up

There are twelve data sets in our experiment. For four citation network data sets (*i.e.,* Citeseer, Cora, Pubmed and Wiki-CS), we follow the previous literature [75] to divide the data sets into a training set, validation set and test set. For eight data sets (*i.e.,* CIFAR10, SVHN, MNIST, Scene15, BBC, Handwritten, Caltech and AWA), we randomly select 10%, 20%, and 30% of the samples as training set. For remaining samples, we select 30% of the samples for the validation

Table 4.1: Classification results (%) of all methods on four citation network data sets (*i.e.,* Citeseer, Cora, Pubmed and Wiki-CS).

Methods	Citeseer	Cora	Pubmed	Wiki-CS
GCN	70.50 ± 0.26	81.60 ± 0.48	79.20 ± 0.32	76.80 ± 0.12
GAT	72.60 ± 0.21	83.18 ± 0.15	79.14 ± 0.87	77.11 ± 0.11
GLCN	72.50 ± 0.68	85.26 ± 0.58	78.80 ± 0.33	77.39 ± 0.23
JLGCN	73.58 ± 0.54	83.77 ± 0.21	79.32 ± 0.84	77.88 ± 0.13
DIAL-GNN	73.89 ± 0.50	84.50 ± 0.39	79.31 ± 0.30	77.64 ± 0.15
M-GLCN	74.15 ± 0.18	84.25 ± 0.34	79.11 ± 0.26	76.13 ± 0.32
GLR-GNN	74.22 ± 0.23	84.75 ± 0.39	79.31 ± 0.14	78.75 ± 0.28
Proposed	**74.85** ± 0.42	**85.77** ± 0.52	**79.84** ± 0.29	**79.36** ± 0.18

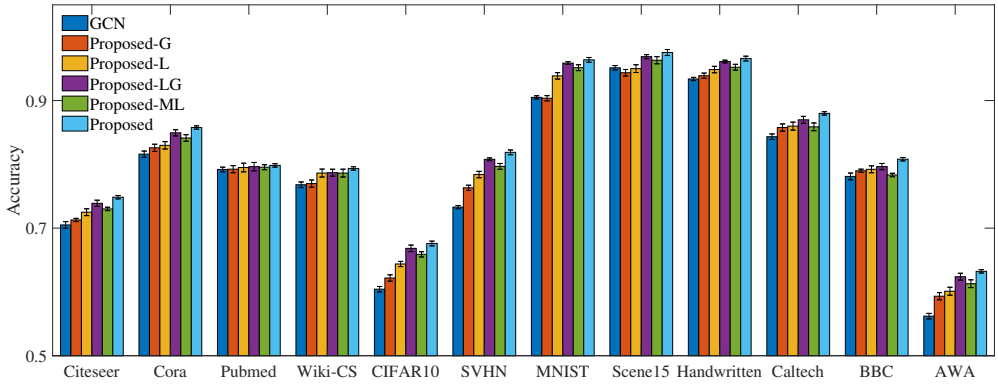
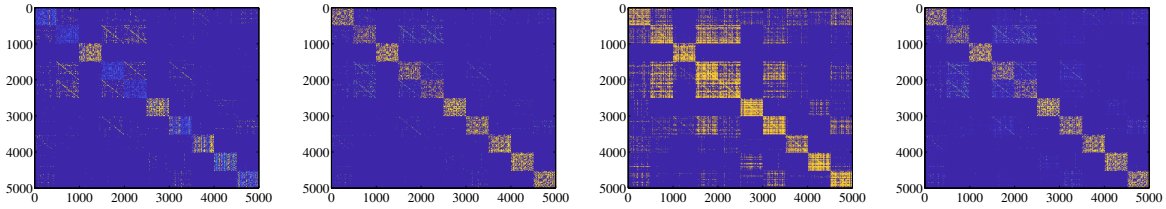


Figure 4.4: Classification accuracy of our methods and GCN on all twelve data sets.



(a) Original graph (b) Local graph (c) Global graph (d) Fused graph

Figure 4.5: Visualization of four types of graphs (*i.e.,* original graph, local graph, global graph, and fused graph) .

set and 30% of the samples for test set. All methods were verified by the ten-fold cross-validation scheme. Moreover, we repeated this scheme ten times and reported the averaged results and the corresponding standard deviations of 100 experiments as the final result. For the model selection, we referred to the corresponding literature to make them output the best performance. In our experiment, our method was set 2 hidden layers, each of which had 30 units. We set the maximum of epochs as 5000 for the training process with the Adam optimizer, and stopped the training process if the loss did not decrease for 100 consecutive epochs. In addition, we set both the initial learning rate and the weight decay as 0.005. For seven image data sets and a text data set, we constructed the initial graph by a kNN graph (*i.e.,* $k = 10$), and set the k value as 5, 10, and 15, to obtain multiple graphs. Moreover, the network weights of all methods were initialized by Glorot initialization.

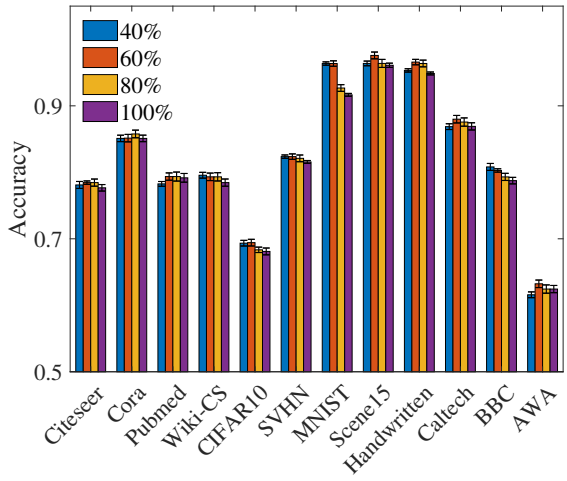


Figure 4.6: Classification accuracy of our methods and Proposed-RO on six data sets.

Table 4.2: Detail of the five proposed methods ($\times$ indicates that the method does not contain this part, $\surd$ means that the method contains this part).

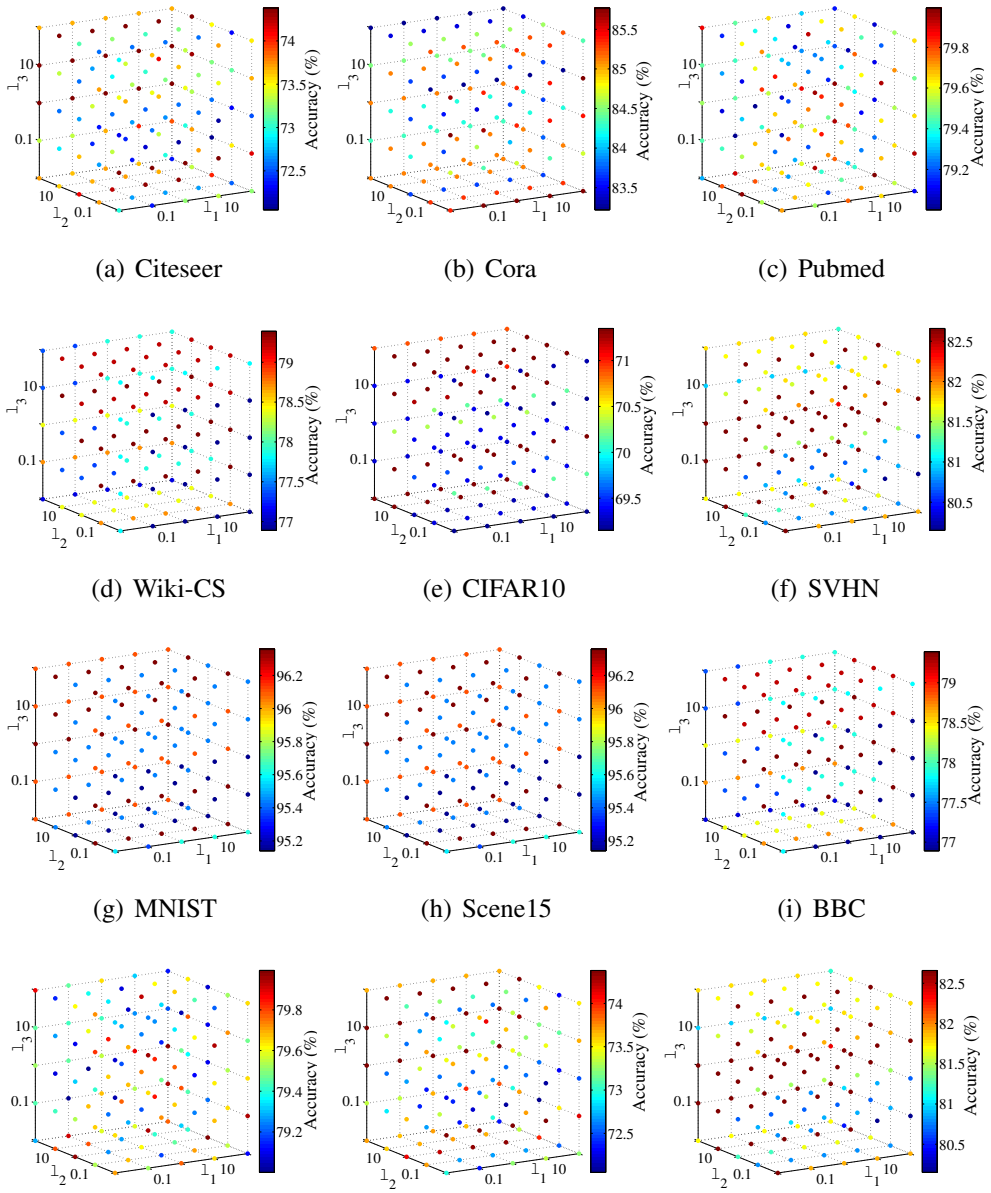
Methods	Global learning	Local learning	Multi-graph learning
Proposed-L	$\surd$	$\times$	$\times$
Proposed-G	$\times$	$\surd$	$\times$
Proposed-LG	$\surd$	$\surd$	$\times$
Proposed-ML	$\surd$	$\times$	$\surd$
Proposed	$\surd$	$\surd$	$\surd$

4.4.2 Result analysis

We report the results of four citation network data sets in Table 4.1 and the results of eight data sets in Figure 4.3.

First, our method achieved the best performance, followed by GLR-GNN, DIAL-GNN, M-GLCN, GLCN, JLGCN, GAT and GCN. For example, in Table 4.1, our method improved by on average 0.63%, 1.02%, 0.53%, 0.61% compared to the best comparison method GLR-GNN, in terms of classification accuracy. In Figure 4.3, our method improved by on average 0.89%, 1.17%, 1.56%, compared to the comparison method GLR-GNN, in terms of different label ratio, *i.e.,* 10%, 20%, 30%, on eight data sets. The possible reason is that our method considers both multi-graph learning and representation learning. As a result, each graph provides the complementary information different from other graphs to find the edges excluded in some graphs and all graphs provide the common information to remove the noisy edges only existed in a small part of the graphs.

Second, both our method and M-GLCN are designed to learn the graph structure of the data from multiple graphs, while the other five methods directly adjust an initial graph. As a result, the multi-graph methods (*i.e.,* our method and M-GLCN) outperformed others (*i.e.,* GCN, GAT, GLCN, JLGCN, DIAL-GNN and GLR-GNN). For example, the multi-graph methods improved by on average 2.35%, compare to others, on eight data sets with different label ratios. This implies the feasibility to take into account the multi-graph learning for graph learning in deep



(a) Citeseer (b) Cora (c) Pubmed

(d) Wiki-CS (e) CIFAR10 (f) SVHN

(g) MNIST (h) Scene15 (i) BBC

(j) Handwritten (k) Caltech (l) AWA

Figure 4.7: Classification accuracy of our method at different parameter settings (*i.e.,* λ_1, λ_2, and λ_3) on twelve data sets.

learning models. In addition, our proposed method is superior to M-GLCN. The possible reason is that our method considers both the global structure and the local structure of the data by a regularization term.

Third, the DGCN methods (*i.e.,* our method, DIAL-GNN, GLCN, GLR-GNN, M-GLCN and JLGCN) beat GCN. For example, these methods improved by on average 4.89%, 4.36%, 3.69%, 4.68%, 4.51 and 3.38%, respectively, compared to GCN, on all data sets. This clearly demon-

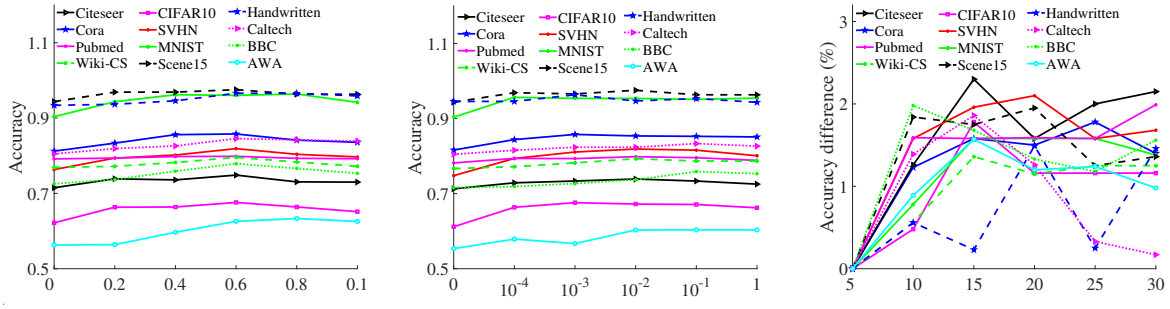


Figure 4.8: Results of our method at different variations of parameter η (left), parameter β (middle), and parameter k (right) .

strates that dynamic graph learning is helpful to improve the performance of GCN methods.

4.4.3 Ablation study

4.4.3.1 Sensitive analysis of different graphs

Our method learns multiple graphs from low-dimensional space of the original data to preserve their local structures and global structure. In this section, we verified the effectiveness of different graphs by extending our method in Eq. (4.9) to the following methods, and list the detail of five proposed methods in the Table 4.2.

- **Proposed-L** considers the local structure in one graph by removing the second term in Eq. (4.9) and setting $k = 5$.
- **Proposed-G** considers the global structure by removing the first term in Eq. (4.9).
- **Proposed-LG** considers the local structure in one graph by setting $k = 5$ and the global structure.
- **Proposed-ML** considers the local structure in multiple graphs by removing the second term in Eq. (4.9) and setting $k = [5, 10, 15]$.

We reported the classification results of six methods (*i.e.,* Proposed-L, Proposed-G, Proposed-ML, Proposed-LG, GCN and Proposed) in Figure 4.4. It is noteworthy that the results of both Proposed and GCN are same as them in Figure 4.3 and Table 4.1. First, Proposed-L outperformed Proposed-G, improving by on average 1.35% on twelve data sets. This shows that the local structure is more useful than the global structure, similar to the conclusion in [91]. However, Proposed-G beat GCN on all data sets, implying the effectiveness of dynamic graph learning and the global structure preservation. Second, Proposed-ML improved by on average 1.14%, compared to Proposed-L, on twelve data sets. It contributes to the fact that a single graph often cannot capture all the information of the data [110]. This clearly demonstrates the feasibility of multi-graph learning in our method. Third, Proposed outperformed Proposed-LG as the latter only uses a single original graph and Proposed uses multiple graphs. This verifies that the learning with multiple graphs may be better than the learning with the single graph [110].

4.4.3.2 Visualization of different graph

In this subsection, we compare the difference among four graphs (*i.e.,* the local graph, the global graph, the fused graph, and the original kNN graph) on the data set MNIST, where we set k = 200 and visualized the results in Figure 4.5.

From Figure 4.5, we have the observations as follows. First, either the global graph or the local graph has more clear structures, compared to the original kNN graph. This indicates the effectiveness of either local structure learning or the global structure learning, as shown in the literature of traditional graph learning [188]. Second, the local graph is more clear, compared to the structure of the global graph. This shows that local information is more important than the global structure in classification tasks. Third, the fused graph is the most clear one among all the graphs. This shows that it is reasonable for our method to consider both the local structure and the global structure.

4.4.3.3 Effectiveness of the dimensionality reduction

In this section, we conduct experiment to verify the effectiveness of dimensionality reduction of our method by varying the value of d' in $\mathbf{\Theta} \in \mathbb{R}^{d \times d'}$. We report the classification accuracy of the proposed method with different values of $d' \in \{40\% * d, 60\% * d, 80\% * d, 100\% * d\}$ in Figure 4.6.

From the results, the proposed method achieves the worst classification results on twelve data sets with the setting of $d' = 100\% * d$. This clearly demonstrates that (1) dimensionality reduction is effective in our method (2) it is feasible to learn the local structure of the data from a low-dimensional space, as redundant features in the original high-dimensional space may influence the model robustness.

4.4.4 Parameter sensitivity analysis

We evaluated the effectiveness of our method at different settings of the parameters, *i.e.,* λ_1, λ_2, and λ_3 in Eq. (4.9), η in Eq. (4.11), and β in Eq. (4.13).

We summarized the classification results of our method with different settings of the parameters (*e.g.,* λ_1, λ_2, and $\lambda_3 \in \{10^{-2}, 10^{-1}, ..., 10^{2}\}$) on twelve data sets in Figure 4.7, in which color represents the classification accuracy and the color bar stands for the range of variation of classification accuracy with different parameter settings. Obviously, our method is sensitive to the settings of three parameters as different parameter settings made different accuracies. Specifically, the variation ranges of accuracy were about 2.45%, 2.38%, 0.84%, 2.10% 1.88%, 2.17%, 1.19%, 1.08%, 2.04%, 1.89%, 2.45% and 1.88% on data sets Citeseer, Cora, Pubmed, Wiki-CS, CIFAR10, SVHN, MNIST, Scene15, BBC, Handwritten, Caltech and AWA respectively. This shows (1) our method can learn useful information from the original graph; (2) the sparsity of $\hat{\mathbf{A}}^{(0)}$ may affect the performance of the model; and (3) our method easily achieves good performance by setting the ranges of λ_1, λ_2, and λ_3 as [0.1, 10].

The parameter η in Eq. (4.11) conducts the trade-off between the local structure graph $\bar{\mathbf{A}}$ and the global structure graph $\hat{\mathbf{A}}^{(0)}$. We reported the classification results of our method with different settings of the parameter η on the left sub-figure of Figure 4.8. The larger the value of η, the less importance the global structure is. Specifically, the accuracy of our method increased with the varied values of η from 0 to 0.6, and decreased with the varied values of η from 0.6 to 1. This indicates that (1) the local structure is more important than the global structure in our method and (2) our method needs both of them because either the complementary information in the individual graph or the common information among them may improve the quality of the graph in our method.

We reported the classification accuracy of our method with the variations of the parameter β in the middle sub-figure of Figure 4.8. Clearly, if β was set to 0, the performance of our method is similar to GCN. However, the accuracy increased while the values of β were varied from 10^{-4} to 1. This verifies that the regularization term is important in our method. Hence, the graph learning regularization term is essential in our method, similar to the conclusion in the literature of the DGCN methods [104, 133].

The right sub-figure of Figure 4.8 reports the accuracy difference between the accuracy of our method under different k settings (*e.g.,* $k \in \{5, 10, 15, 20, 25, 30\}$) and the classification accuracy of our method in the case $k = 5$. Based on the experimental results, the accuracy of our method increased with the varied values of k from 5 to 15, and decreased with the varied values of k from 15 to 30. In addition, our method achieved the best experimental results while the values of k is either 15 or 20. This demonstrates that (1) the small values of k cannot fully describe the neighborhoods of the data and large values of k will increase the chance of wrong neighborhoods which makes the relationship among samples less discriminative, and (2) our proposed local structure learning method is sensitive to the value of k.

4.5 Conclusion

In this chapter, we have designed a novel dynamic GCN by proposing a multi-graph fusion method to improve the quality of the graph in the GCN method. To do this, our method first learnt the unified local graph preserving the local structure from multiple initial graphs and the global graph preserving the global structure from the low-dimensional space of the original high-dimensional data, and then proposed a novel fusion method to integrate their complementary information and common information, aiming at correctly capturing the intrinsic graph structure inherent in the data. Experimental results on twelve real data sets showed the effectiveness of our method, compared to state-of-the-art comparison methods.

This chapter has been published in the CORE rank A* journal, *i.e.,* IEEE Transactions on Neural Networks and Learning Systems [36].

Chapter 5

Multi-view Unsupervised Graph Representation Learning

5.1 Introduction

Unsupervised graph representation learning (UGRL) can output discriminative representations using both structural (*e.g.,* the relationship of data points) and non-structured (*e.g.,* node feature) information of the data, making it easily extract useful information to benefit downstream tasks [146]. To achieve this, the objective function of the UGRL is often designed to maximize mutual information (MI) between the input and its related information. However, the MI maximization is popular transferred to the Jensen-Shannon divergence maximization, as its computation cost is expensive [90]. This results in contrastive learning based UGRL (CL-UGRL). Recently, CL-UGRL shows its superiority in unsupervised representation learning, and has been widely used in various learning tasks, such as clustering and community detection [58].

The CL-UGRL method mainly includes three key components, *i.e.,* data augmentation, graph convolutional network (GCN) encoder, and contrastive loss. Specifically, data augmentation aims at creating rational data for contrastive learning by applying certain transformation [107]. GCN encodes the topology structure and node features of the data. The contrastive loss is related to the definitions of anchors, positive samples and negative samples. Moreover, it pushes the anchor embedding similar to positive emedding and dissimilar to negative embedding [137]. Although current CL-UGRL methods have achieved success, due to the complexity of the data, previous CL-UGRL methods still have many limitations.

The commonly used graph data augmentation methods are random perturbation, including node dropping, feature masking, subgraph and edge perturbation. For example, to conduct data augmentation, [183] employed node dropping and subgraph, while [173] employed edge perturbation and feature masking. However, random perturbation might destroy the inherent structure and property of the graph, affecting the model performance. To address this issue, [50] used the diffusion method to generate augmented data, which avoids the destruction of the data struc-

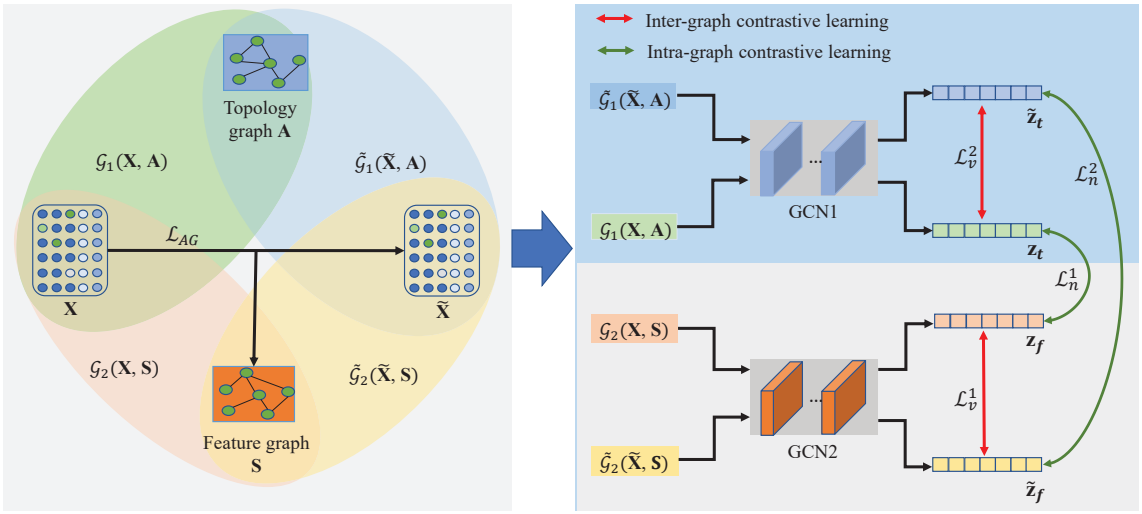


Figure 5.1: The flowchart of our method. It first proposes an adaptive data augmentation to generate multi-view information, *i.e.,* $\mathcal{G}_1 = (\mathbf{X}, \mathbf{A})$, $\widetilde{\mathcal{G}}_1 = (\widetilde{\mathbf{X}}, \mathbf{A})$, $\mathcal{G}_2 = (\mathbf{X}, \mathbf{S})$ and $\widetilde{\mathcal{G}}_2 = (\widetilde{\mathbf{X}}, \mathbf{S})$, and then designs two types of contrastive losses, *i.e.,* intra-graph contrastive loss and inter-graph contrastive loss, for conducting multi-view contrastive learning.

ture. However, such a method is independent of contrastive learning, and thus fails to adaptively consider the impacts of input graphs.

Compared to unsupervised representation learning methods, the UGRL maintains the graph structure of data points during representation learning, thereby outputting more discriminative representations [44]. Therefore, the quality of the graph is very important to the performance of the UGRL method. In current UGRL methods, the topology graph and the feature graph are two popular graph structures to represent the relationship between data points. The topology graph usually comes from the real world to reflect the connection of data points, such as social networks and citation networks [168]. The feature graph is constructed by the kNN method or the ϵ-graph method to represent the similarity of data points in the feature space [92]. These two kinds of graphs contain different information, which may provide complementary inforamtion to benefit the UGRL. However, existing UGRL usually cannot take full advantage of the information. Therefore, the fusion of two graphs is a way to improve the UGRL.

In this chapter, we propose a multi-view CL-UGRL method to address the aforementioned issues by two key components, *i.e.,* adaptive data augmentation and multi-view contrastive learning. Our adaptive data augmentation first generates the feature graph from the feature space and then designs a deep graph learning model to jointly update the feature graph and the new representation based on the original representation. We further combine the feature and topology graphs with the original and new representations to form multi-view information, which is fed into two GCNs to generate multi-view embedding features. Two kinds of contrastive losses are designed on multi-view embedding features to explore the complementary information between the topology and feature graphs. Different from traditional data augmentation independent on contrastive learning, we embed adaptive data augmentation and multi-view contrastive learning in a frame-

work to form an end-to-end model, which jointly conducts data augmentation and multi-view contrastive learning. As a result, data augmentation is iteratively updated by the adjusted multi-view contrastive learning and vice versa.

Compared to previous methods, the main contributions of our method are summarized as follows. First, we propose a new data augmentation method, which generates a new graph and new representation, aiming at preserving the intrinsic structure of the data to generate multi-view information for contrastive learning. Furthermore, it is dependent on our multi-view contrastive learning as both of them are embedded in the same framework. Second, we propose a new multi-view contrastive learning framework, including two kinds of contrastive losses, which provide complementary information between the feature and topology graphs to strength multi-view contrastive learning. Third, we embed adaptive data augmentation and multi-view contrastive learning in a unified framework so that each of them can be iteratively adjusted by the other's update. Extensive experiments on benchmark data sets clearly demonstrate that our method outperforms the state-of-the-art methods on different downstream tasks.

5.2 Related work

5.2.1 Unsupervised graph representation learning

Unsupervised graph representation learning (UGRL) aims to first learn a low-dimension latent representation of nodes, and then use the learned representations for downstream graph analysis tasks. Traditional UGRL methods including matrix factorization methods and random walk methods. Matrix factorization methods obtain a low-dimensional node representation by singular value decomposition of relational matrix or similar matrix, such as Graph Factorization [3], Grarep [13] and Hope [102]. Random walk methods generate a random walk sequence as the input feature, such as Word2vec [97] and node2vec [46].

With the rapid development of graph neural networks (GNN), a large number of graph representation learning methods based on GNN have been proposed to exhibit strong effectiveness, compared to traditional methods. For example, GraphSAGE [48] trains GNN by a random-walk based objective in its unsupervised setting. Other typical methods have SDNE [145] and DNGR [14]. Recently, DGI [144] applies the idea of mutual information maximization to the graph domain and obtains the strong performance in an unsupervised pattern.

5.2.2 Contrastive learning

Contrastive learning methods aim to learn discriminative representations by contrasting positive and negative samples. Therefore, data augmentation and contrastive loss are two crucial component in contrastive learning.

The popular data augmentation methods have node dropping, edge perturbation, attribute masking and subgraph [173]. These methods generate augmented data by randomly disturbing the

features or structures of graph data. Due to the lack of priori information, they usually has limitations. Recently, [50] employ graph diffusion to generate augmentation graph and [198] design an adaptive data augmentation method. DGI [144] applies the InfoMax principle [53] on graph data by contrasting the representations of the node-level representations and the graph-level representations for node classification tasks. Different from DGI, InfoGraph [125] aims at node classification tasks by contrasting the representations of the graph-level representations and the substructure-level representations with different granularity.

5.3 Method

In this chapter, $\mathcal{G}_1 = (\mathbf{X}, \mathbf{A})$ consists of $\mathbf{A}$ and $\mathbf{X}$, where $\mathbf{X} \in \mathbb{R}^{N \times d}$ and $\mathbf{A} \in \mathbb{R}^{N \times N}$ represent the feature matrix and the topology graph, respectively. N is the number of nodes and d is the dimensions of node features. Moreover, $a_{ij} = 1$ indicates that there is an edge between the i-th node and the j-th node, otherwise $a_{ij} = 0$.

The proposed framework is shown in Figure 5.1. Its key components include adaptive data augmentation and multiple contrastive learning. Specifically, given $\mathbf{X}$ and $\mathbf{A}$, our adaptive data augmentation adaptively learns both the new representations $\widetilde{\mathbf{X}}$ and the feature graph $\mathbf{S}$ of $\mathbf{X}$, followed by generating multi-view information, *i.e.,* $\mathcal{G}_1 = (\mathbf{X}, \mathbf{A})$, $\widetilde{\mathcal{G}}_1 = (\widetilde{\mathbf{X}}, \mathbf{A})$, $\mathcal{G}_2 = (\mathbf{X}, \mathbf{S})$ and $\widetilde{\mathcal{G}}_2 = (\widetilde{\mathbf{X}}, \mathbf{S})$. In particular, two-view information with the same graph, *e.g.,* $\mathcal{G}_1$ and $\widetilde{\mathcal{G}}_1$ (or $\mathcal{G}_2$ and $\widetilde{\mathcal{G}}_2$), is input into the same encoder to output new embedding features $\mathbf{Z}_t$ and $\widetilde{\mathbf{Z}}_t$ (or $\mathbf{Z}_f$ and $\widetilde{\mathbf{Z}}_f$), respectively. Based on resulted multi-view embedding features, we further design the intra-graph contrastive loss and the inter-graph contrastive loss to conduct multi-view contrastive learning.

5.3.1 Adaptive data augmentation

Existing data augmentation strategies are popularly used for Euclidean data, but are difficult for the graph data [20]. In the literature, graph contrastive learning usually relies on the contrast between node embedding features in different views [109]. To achieve this, data augmentation usually corrupts original data structures to obtain multi-view information by random perturbation, including node dropping, edge perturbation, feature masking, *etc.* However, random perturbation might destroy the intrinsic structure of the data. For example, either the removal or addition of edges might drastically change the identity or even the compound validity of bio-molecule data [173]. Thus, data augmentation should preserve the intrinsic structures of the data, which can help learnt embedding features insensitive to the perturbation on unimportant nodes and edges. Additionally, previous data augmentation is independent on contrastive learning. If these two processes are integrated in the same framework, each of them can be alternatively adjusted by the other. In this way, the weakness of data augmentation can further be improved by contrastive learning, while the updated data augmentation can adjust contrastive learning. However, few literature focused on this.

In this chapter, we develop a novel adaptive data augmentation method to dynamically preserve

the intrinsic structures of the data in this section and then integrates it with multi-view contrastive learning in a unified framework (Section 5.3.4). More specifically, the adaptive data augmentation includes two parts, *i.e.,* node feature augmentation and graph structure augmentation, while multi-view contrastive learning is used to explore complementary information between the feature and topology graphs.

5.3.1.1 Node feature augmentation

The method of randomly masking node features recovers masked node features using their remaining features, and is a widely used method of data augmentation [107]. However, it might destroy the intrinsic structure of the data, thereby affecting the generation capability of the model. Hence, an effective data augmentation method is demand.

In traditional machine learning methods, feature selection is widely used to remove redundant features. For example, [99] employed a joint $\ell_{2,1}$-norm regularization to select important features while [154] utilized sparse subspace learning to learn feature weights. Motivated by this, we propose to design a new data augmentation method to keep significant feature unchanged while perturbing trivial features. Specifically, we propose to learn the significance of features and mask trivial features. To achieve this, we first define the following objective function:

$$\min_{\mathbf{P}} \ \|\mathbf{X} - \mathbf{X}\mathbf{P}\|_F^2 + \|\mathbf{P}\|_F^2 \tag{5.1}$$

where $\mathbf{P} \in \mathbb{R}^{d\times d}$ is the weight matrix of features. We focus on corrupting the original feature graph at its feature levels. Specifically, we define an indicator vector $\mathbf{m} \in \{0,1\}^{d\times 1}$ to assign zeros to the features with the least weights in the weight matrix $\mathbf{P}$. As a result, the generated new node representation $\widetilde{\mathbf{X}}$ by the proposed node feature generation is:

$$\widetilde{\mathbf{X}} = [\mathbf{x}_1 \circ \mathbf{m}, ..., \mathbf{x}_N \circ \mathbf{m}] \tag{5.2}$$

where $\circ$ is the element-wise multiplication.

5.3.1.2 Graph structure augmentation

Edge perturbation perturbs the connectivity of the graph by randomly adding or dropping certain ratio of edges. However, deleting useful edges will reduce the discriminative ability of embedding features, and thus it has been proven to be unfriendly to biological and chemical data [173]. Additionally, the graph structure is invariant during the process of data augmentation. However, the feature graph $\mathbf{S}$ is constructed by the original representations $\mathbf{X}$, which might contain noise or redundancy to affect the quality of $\mathbf{S}$. Eq. (5.1) is used to output better representation than $\mathbf{X}$, while the new representation can be used to further update $\mathbf{S}$, so that the updated $\mathbf{S}$ can really reflect the intrinsic structure of the data. Graph learning has been designed to dynamically adjust the graph structure, thereby learning the intrinsic structure of the data. For example, [191] designed to use dynamic graph learning for unsupervised feature selection while [23] proposed to embed representation learning with graph learning in a unified framework. Inspired by this, we design a graph learning method to highlight the intrinsic structures of the data as follows.

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i,j=1}^{N} \|\mathbf{x}_i - \mathbf{x}_j\| s_{ij} + \lambda \|\mathbf{S}\|_F^2 \\ &\forall i, \mathbf{s}_i \mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0 \end{aligned} \tag{5.3}$$

where $\mathbf{1}$ represents the all-one-element vector and λ is a tuning parameter. In this work, Eq. (5.1) and Eq. (5.3) update the weight matrix $\mathbf{P}$ and the graph $\mathbf{S}$, respectively. Hence, we integrate Eq. (5.1) and Eq. (5.3) in a unified framework as follows:

$$\begin{aligned} &\mathcal{L}_{AG} = \|\mathbf{X} - \mathbf{X}\mathbf{P}\|_F^2 + \lambda_1 \sum_{i,j=1}^{N} \|\mathbf{x}_i\mathbf{P} - \mathbf{x}_j\mathbf{P}\| s_{ij} + \lambda_2 \|\mathbf{S}\|_F^2 \\ &\forall i, \mathbf{s}_i \mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0 \end{aligned} \tag{5.4}$$

where λ_1 and λ_2 are tuning parameters and $\mathbf{P}$ is a trainable parameter. Based on [62], we update $\mathbf{S}$ by:

$$s_{ij} = \frac{exp(ReLU(\mathbf{q}^T|\mathbf{x}_i\mathbf{P} - \mathbf{x}_j\mathbf{P}|))}{\sum_{j=1}^{n} exp(ReLU(\mathbf{q}^T|\mathbf{x}_i\mathbf{P} - \mathbf{x}_j\mathbf{P}|))} \tag{5.5}$$

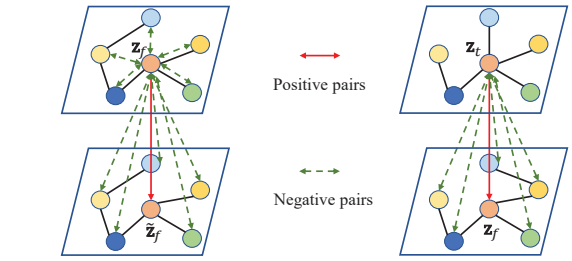
where $ReLU(.)$ is the activation function and $\mathbf{q} \in \mathbb{R}^{h \times 1}$ is a trainable parameter.

Given the new representation $\widetilde{\mathbf{X}}$ and feature graph $\mathbf{S}$, multiple-view information (*i.e.,* $\widetilde{\mathcal{G}}_1 = (\widetilde{\mathbf{X}}, \mathbf{A})$, $\mathcal{G}_2 = (\mathbf{X}, \mathbf{S})$ and $\widetilde{\mathcal{G}}_2 = (\widetilde{\mathbf{X}}, \mathbf{S})$) is generated by our proposed adaptive data augmentation method. Different from previous data augmentation, our method iteratively updates both $\mathbf{S}$ and $\widetilde{\mathbf{X}}$, thereby avoiding the drawbacks of previous methods, *e.g.,* sensitive to perturbation on trivial nodes and edges, and destroying the structures of the data.

5.3.2 GCN encoder

Given multi-view information (*i.e.,* $\mathcal{G}_1$, $\mathcal{G}_2$, $\widetilde{\mathcal{G}}_1$ and $\widetilde{\mathcal{G}}_2$), two GCN encoders are utilized to generate multi-view embedding features as different graphs might provide complementary information to each other [71]. For example, the topology graph $\mathbf{A}$ contains the similarity from the real world while the feature graph $\mathbf{S}$ gathers the similarity from the feature space. Specifically, given the topology graph $\mathbf{A}$, we input both $\mathcal{G}_1$ and $\widetilde{\mathcal{G}}_1$ into GCN1 to output their corresponding embedding features $\mathbf{Z}_t$ and $\widetilde{\mathbf{Z}}_t$, aiming at simultaneously updating the new representation $\widetilde{\mathbf{X}}$ and the embedding features. In particular, GCN1 is used for both $\mathcal{G}_1$ and $\widetilde{\mathcal{G}}_1$ as they share the same topology graph, with which it is easy to output a high quality $\widetilde{\mathbf{X}}$. Similarly, we input both $\mathcal{G}_2$ and $\widetilde{\mathcal{G}}_2$ into GCN2 to output embedding features $\mathbf{Z}_f$ and $\widetilde{\mathbf{Z}}_f$, as well as to update both $\widetilde{\mathbf{X}}$ and $\mathbf{S}$.

$$\begin{cases} \mathbf{Z}_f = GCN2(\mathbf{X}, \mathbf{A}) = MLP(\hat{\mathbf{A}}\sigma(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}_f^{(0)})\mathbf{W}_f^{(1)}) \\ \mathbf{Z}_t = GCN1(\mathbf{X}, \mathbf{S}) = MLP(\hat{\mathbf{S}}\sigma(\hat{\mathbf{S}}\mathbf{X}\mathbf{W}_t^{(0)})\mathbf{W}_t^{(1)}) \end{cases} \tag{5.6}$$



(a) Intra-graph contrastive learning (b) Inter-graph contrastive learning

Figure 5.2: Illustration of the difference between the intra-graph contrastive loss and inter-graph contrastive loss. Specifically, both of them have the same definition for positive embeddings, but are with different definitions for negative embeddings.

$$\begin{cases} \widetilde{\mathbf{Z}}_f = GCN2(\widetilde{\mathbf{X}}, \mathbf{A}) = MLP(\hat{\mathbf{A}}\sigma(\hat{\mathbf{A}}\widetilde{\mathbf{X}}\mathbf{W}_f^{(0)})\mathbf{W}_f^{(1)}) \\ \widetilde{\mathbf{Z}}_t = GCN1(\widetilde{\mathbf{X}}, \mathbf{S}) = MLP(\hat{\mathbf{S}}\sigma(\hat{\mathbf{S}}\widetilde{\mathbf{X}}\mathbf{W}_t^{(0)})\mathbf{W}_t^{(1)}) \end{cases} \tag{5.7}$$

where $\hat{\mathbf{A}} = {\mathbf{D}_\mathbf{A}}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I}){\mathbf{D}_\mathbf{A}}^{-\frac{1}{2}}$, $\hat{\mathbf{S}} = {\mathbf{D}_\mathbf{S}}^{-\frac{1}{2}}(\mathbf{S}+\mathbf{I}){\mathbf{D}_\mathbf{S}}^{-\frac{1}{2}}$, $\mathbf{D}_\mathbf{A}$ and $\mathbf{D}_\mathbf{S}$ are the degree matrix of $\mathbf{A}+\mathbf{I}$ and $\mathbf{S}+\mathbf{I}$, respectively, $MLP(.)$ represents the Multi-Layer Perception.

5.3.3 Multi-view contrastive learning

The key idea of contrastive learning is to define semantically similar (positive) and dissimilar (negative) pairs, encouraging the embedding features of similar pairs $(\mathbf{x}, \mathbf{x}^+)$ to be close, and those of dissimilar pairs $(\mathbf{x}, \mathbf{x}^-)$ to be far away. Specifically, the contrastive learning is to achieve the following objective:

$$sim(f(\mathbf{x}), f(\mathbf{x}^+)) \gg sim(f(\mathbf{x}), f(\mathbf{x}^-)) \tag{5.8}$$

where $f(.)$ represents encoder and $sim(.)$ measures the similarity of embedding features of two nodes. As our proposed adaptive data augmentation generates multi-view information, we design two kinds of contrastive losses (*i.e.,* intra-graph contrastive loss and inter-graph contrastive loss) to explore rich contrastive relations among embedding features, *i.e.,* $\mathbf{Z}_f$, $\mathbf{Z}_t$, $\widetilde{\mathbf{Z}}_f$, and $\widetilde{\mathbf{Z}}_t$.

To do this, we employ two GCNs to generate multi-view embedding features. A intra-graph contrastive loss for the embedding features derived from the same encoder is then built, aiming at separating positive embeddings from negative embeddings within the same graph structure. Meanwhile, an inter-graph contrastive loss for the embedding features from different graph structures can also be built, aiming to separate positive embeddings from negative embeddings across graphs. Figure 5.2 illustrates the difference of two kinds of contrastive losses.

5.3.3.1 Inter-graph contrastive learning

The inter-graph contrastive learning is to contrast the embedding features from different graph strucutres, *i.e.,* the topology graph and the feature graph. Specifically, an inter-graph contrastive

loss is designed to pull the embedding features of the same node from different views across graph structures close while pushing the embedding features of one node far away from the embedding features of other nodes from different views across graph structures, *i.e.,*

$$\mathcal{L}_n^1(v_i) = -log\frac{exp(sim(\mathbf{z}_f^{v_i}, \mathbf{z}_t^{v_i}))}{\sum_{j=1}^{N} exp(sim(\mathbf{z}_f^{v_i}, \mathbf{z}_t^{v_j}))} \tag{5.9}$$

$$\mathcal{L}_n^2(v_i) = -log\frac{exp(sim(\widetilde{\mathbf{z}}_f^{v_i}, \widetilde{\mathbf{z}}_t^{v_i}))}{\sum_{j=1}^{N} exp(sim(\widetilde{\mathbf{z}}_f^{v_i}, \widetilde{\mathbf{z}}_t^{v_j}))} \tag{5.10}$$

where v_i represents the i-th node, $sim(a, b) = exp(\frac{a^T b}{\|a\| \cdot \|b\|})/\vartheta$, ϑ represents the temperature factor that controls the concentration level of the distribution. Besides, $\mathbf{z}_f^{v_i} \in \mathbf{Z}_f$, $\mathbf{z}_t^{v_i} \in \mathbf{Z}_t$, and $\widetilde{\mathbf{z}}_f^{v_i} \in \mathbf{Z}_f$, $\widetilde{\mathbf{z}}_t^{v_i} \in \mathbf{Z}_t$. Combining the above two losses, the inter-graph contrastive loss is defined as follows:

$$\mathcal{L}_n = \frac{1}{2N} \sum_{i=1}^{N} (\mathcal{L}_n^1(v_i) + \mathcal{L}_n^2(v_i)) \tag{5.11}$$

5.3.3.2 Intra-graph contrastive learning

Intra-graph contrastive loss is used to build the contrast from the same graph structure, which is popular in CL-UGRL [165]. In this chapter, the intra-graph contrastive loss is designed to pull the embedding features of the same node from different views in the same graph structure close while pushing the embedding features of one node far away from the embedding features of other nodes from both the same view and different views, which come from the same graph structure, *i.e.,*

$$\mathcal{L}_v^1(v_i) = -log\frac{exp(sim(\mathbf{z}_f^{v_i}, \widetilde{\mathbf{z}}_f^{v_i}))}{\sum_{j=1}^{N} exp(sim(\mathbf{z}_f^{v_i}, \widetilde{\mathbf{z}}_f^{v_j}))} \tag{5.12}$$

However, the node shares the same graph structure but is with different views, which might cause the embedding features in different views similar by Eq. (5.12). To make the embedding features of the same node in different views distinguishable, we consider adding the diversity constraints into the intra-graph contrasitive loss.

$$\mathcal{L}_v^1(v_i) = -log\frac{exp(sim(\mathbf{z}_f^{v_i}, \widetilde{\mathbf{z}}_f^{v_i}))}{\sum_{j=1}^{N} exp(sim(\mathbf{z}_f^{v_i}, \widetilde{\mathbf{z}}_f^{v_j})) + \Theta} \tag{5.13}$$

where $\Theta = \sum_{j=1}^{N} exp(sim(\mathbf{z}_f^{v_i}, \mathbf{z}_f^{v_j}))$. Similarly, we define $\mathcal{L}_v^2(v_i)$ between $\mathcal{G}_2$ and $\widetilde{\mathcal{G}}_2$ as:

$$\mathcal{L}_v^2(v_i) = -log\frac{exp(sim(\mathbf{z}_t^{v_i}, \widetilde{\mathbf{z}}_t^{v_i}))}{\sum_{j=1}^{N} exp(sim(\mathbf{z}_t^{v_i}, \widetilde{\mathbf{z}}_t^{v_j})) + \sum_{j=1}^{N} exp(sim(\mathbf{z}_t^{v_i}, \mathbf{z}_t^{v_j}))} \tag{5.14}$$

Combining Eq. (5.13) with Eq. (5.14), the intra-graph contrastive loss $\mathcal{L}_v$ is:

$$\mathcal{L}_v = \frac{1}{2N} \sum_{i=1}^{N} (\mathcal{L}_v^1(v_i) + \mathcal{L}_v^2(v_i)) \tag{5.15}$$

Finally, combining the inter-graph loss with the intra-graph loss, the objective function of the proposed multi-view contrastive learning is defined as follows.

$$\mathcal{L}_{CL} = (1-\eta)\mathcal{L}_v + \mathcal{L}_n \tag{5.16}$$

where η is a non-negative tuning parameter.

The intra-graph contrasitive loss is defined in different views of the same graph structure, while the inter-graph contrastive losses is defined in different views across different graph structures. Hence, Eq. (5.16) considers the individual graph structures (*e.g.,* the feature graph or the topology graph) by inter-graph contrastive losses as well as the connection between two graphs by inter-graph losses, so that it easily obtains complementary information among different graph structures.

Table 5.1: Node classification accuracy (%) of all methods on eight data sets.

Datasets	GCN	GAT	DGI	GMI	DGCRL	MVGRL	Proposed
Citeseer	70.30 ± 0.50	72.20 ± 0.26	71.80 ± 0.45	72.4 ± 0.26	72.53 ± 0.46	72.56 ± 0.50	**72.87** ± 0.40
Cora	81.50± 0.20	83.00± 0.30	82.30± 0.31	83.00± 0.30	83.10± 0.40	82.58 ± 0.50	**83.85**± 0.35
Wiki-CS	77.19± 0.12	77.65± 0.11	73.53± 0.14	74.85± 0.08	74.89± 0.26	77.52± 0.08	**79.56**± 0.14
Computers	86.51± 0.19	86.93 ± 0.29	83.95± 0.47	82.21± 0.31	84.71± 0.19	87.52± 0.24	**87.98**± 0.30
Coauthors	93.03± 0.31	92.31± 0.24	92.15± 0.63	92.68± 0.11	92.45± 0.07	92.11± 0.12	**93.10**± 0.14
DBLP	82.70± 0.18	83.45± 0.36	83.20± 0.10	83.29± 0.15	84.19± 0.17	83.64± 0.24	**85.21**± 0.16
Photo	91.30± 0.22	91.80± 0.15	91.61± 0.17	90.68± 0.22	91.58± 0.18	91.74± 0.07	**93.28**± 0.19
Pubmed	79.00± 0.10	79.10± 0.24	77.90± 0.60	79.90± 0.20	79.60± 0.50	80.10±0.50	**80.25**± 0.35

5.3.4 Overall objective function

In order to train an end-to-end CL-UGRL model, we jointly consider multi-view contrastive learning and adaptive data augmentation to have our final objective function as follows.

$$\mathcal{L} = \beta\mathcal{L}_{AG} + \mathcal{L}_{CL} \tag{5.17}$$

where β is a tuning parameter. As a result, adaptive data augmentation is updated by multi-view contrastive learning and vice versa. Eq. (5.17) can be optimized by the standard gradient descent algorithm. We list the pseudo of our method in Appendix. In the downstream task, the mean function is employed to aggregate embedding features outputted by GCN1 and GCN2, *i.e.,* $\mathbf{Z} = (Mean(\mathbf{Z}_t, \widetilde{\mathbf{Z}}_t)||Mean(\mathbf{Z}_f, \widetilde{\mathbf{Z}}_f))$, where $||$ denotes the concatenation operator.

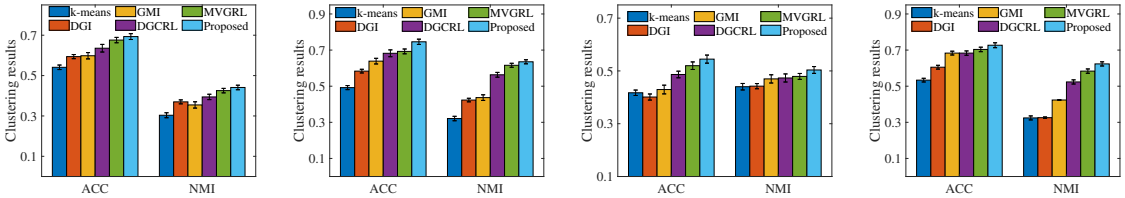
5.4 Experiments

5.4.1 Data sets

The used data sets include citation networks data (*i.e.,* Citeseer, Cora, Pubmed and DBLP), networks data (*i.e.,* Photo and Computers), academic network data (*i.e.,* Coauthors), and reference network data (*i.e.,* Wiki-CS).

Table 5.2: Link prediction performance (%) of all methods on four data sets (*i.e.,* Cora, Citeseer, Pubmed and Photo).

	Cora		Citeseer		PubMed		Photo	
Method	AUC	AP	AUC	AP	AUC	AP	AUC	AP
GCN	92.31 ± 0.40	93.81 ± 0.25	89.14 ± 0.51	90.24 ± 0.51	96.14 ± 0.24	96.20 ± 0.24	92.30 ± 0.20	91.50 ± 0.14
GAT	90.20± 0.75	90.32 ± 0.33	90.80 ± 0.72	90.70 ± 0.61	90.70 ± 0.37	90.10 ± 0.57	91.20 ± 0.52	91.20 ± 0.26
DGI	89.82 ± 0.24	89.70 ± 0.54	95.51 ± 0.21	91.24 ± 0.24	91.35 ± 0.54	92.24 ± 0.42	95.37 ± 0.40	94.60 ± 0.35
GMI	91.24 ± 0.74	91.30± 0.42	92.24 ± 0.54	95.10 ± 0.52	93.87 ± 0.31	94.93 ± 0.34	93.28 ± 0.40	92.80 ± 0.20
DGCRL	94.91 ± 0.41	95.52 ± 0.37	96.12 ± 0.41	95.10 ± 0.37	95.10 ± 0.40	95.00 ± 0.40	93.60 ± 0.37	92.40 ± 0.25
MVGRL	84.92 ± 0.32	89.20 ± 0.48	84.46 ± 0.40	91.52 ± 0.34	92.28 ± 0.40	92.26 ± 0.60	90.05 ± 0.24	89.14 ± 0.34
Proposed	**95.31** ± 0.14	**95.80** ± 0.30	**97.04** ± 0.40	**96.21** ± 0.14	**96.23** ± 0.50	**95.47** ± 0.31	**96.24** ± 0.34	**95.50** ± 0.20



(a) Citeseer (b) Cora (c) Wiki-CS (d) Computers

Figure 5.3: Clustering results, *i.e.,* ACC and NMI, of all methods on four data sets (*i.e.,* Citeseer, Cora, Wiki-CS and Computers).

5.4.2 Comparison methods

The comparison methods include unsupervised learning methods (*i.e.,* Deep Graph Infomax (DGI) [143], Graphical Mutual Information Maximization (GMI) [105], Multi-View Graph Representation Learning (MVGRL) [50] and Deep Graph Contrastive Representation Learning (DGCRL) [197]) and supervised learning methods (*i.e.,* GCN [75] and Graph Attention Networks (GAT) [142]).

5.4.3 Result analysis

5.4.3.1 Node classification

We report the results of node classification of all methods in Table 5.1. Our method achieves the best performance, followed by MVGRL, DGCRL, GMI and DGI, in terms of unsupervised representation learning methods. Specifically, our method improves by on average 0.66%, 1.70%, 2.13%, and 2.45%, compared to four unsupervised comparison methods, in terms of classification accuracy, on eight data sets. Compared to semi-supervised methods (*i.e.,* GCN and GAT) which adopt the label information in the learning process, our method also achieve the superior performance.

5.4.3.2 Clustering

Figure 5.3 illustrates clustering performance of all methods on four datasets, and the performance of other four data sets are in Supplementary Materials. Obviously, our proposed method beats all

Table 5.3: Node classification accuracy of four methods, *i.e.,* Pro-R-F, Pro-R-T, Pro-R-DA and Proposed.

Datasets	Pro-R-F	Pro-R-T	Pro-R-DA	Proposed
Citeseer	72.10±0.20	71.38± 0.40	71.59± 0.37	**72.87**± 0.40
Cora	83.14± 0.25	82.81± 0.20	81.25± 0.18	**83.85**± 0.35
Wiki-CS	79.14± 0.07	78.59± 0.06	78.37± 0.10	**79.56**± 0.14
Computers	86.32± 0.18	87.53± 0.30	85.37± 0.24	**87.98**± 0.30
Coauthors	92.10± 0.15	91.25± 0.07	91.35± 0.12	**93.10**± 0.14
DBLP	84.08± 0.27	83.15± 0.21	83.04± 0.25	**85.21**± 0.16
Photo	91.28± 0.14	92.56± 0.12	91.08± 0.14	**93.28**± 0.19
Pubmed	80.10± 0.13	80.10± 0.15	79.56± 0.31	**80.25**± 0.35

Table 5.4: Classification accuracy of three methods (*i.e.,* Proposed w/o n, Proposed w/o v and Proposed) on eight data sets.

Datasets	Proposed w/o n	Proposed w/o v	Proposed
Citeseer	71.65±0.15	72.10± 0.33	72.87± 0.40
Cora	82.94± 0.18	82.58± 0.04	83.85± 0.35
Wiki-CS	77.59± 0.14	77.58± 0.13	79.56± 0.14
Computers	87.10± 0.14	86.32± 0.24	87.98± 0.30
Coauthors	92.14± 0.15	92.56± 0.07	93.10± 0.14
DBLP	83.26± 0.18	84.53± 0.14	85.21± 0.16
Photo	92.49± 0.09	92.36± 0.12	93.28± 0.19
Pubmed	80.10± 0.13	79.68± 0.13	80.25± 0.35

comparison methods on all datasets. For example, our method improves by on average 2.19%, 2.37%, in terms of ACC and NMI, on four data sets, compare to the best comparison method MVGRL.

5.4.3.3 Link prediction

Table 5.2 illustrates the link prediction results of all methods on four datasets. Our proposed method outperforms all comparison methods. For example, our method improves by on average 1.27% and 1.24%, in terms of AUC and AP, on four data sets, compared to the best comparison method DGCRL.

Finally, our method achieves the best performance in term of three kind of real applications. This demonstrates that our method can output representations that are beneficial for downstream learning tasks. Moreover, it is feasible to generate auxiliary information to mine the information hidden in the graph data through contrastive learning. The reasons is that our adaptive data augmentation method produces important information of the graph data from both the structure and feature levels, while all unsupervised comparison methods adopt the random perturbation method to generate augmented data. Moreover, our method makes full use of the topology and feature information of the graph data.

5.4.4 Ablation Study

5.4.4.1 Effectiveness of data augmentation

To verify the effectiveness of our data augmentation method, we propose three new methods, *i.e.,* Proposed-R-F, Proposed-R-T and Proposed-R-DA, based on our method. Specifically, Proposed-R-F removes the first term of Eq. (5.4), *i.e.,* without considering the adaptive data augmentation of node features. Proposed-R-T removes the second and third terms of Eq. (5.4), *i.e.,* without considering the adaptive data augmentation of the graph structure. Proposed-R-DA employs random perturbation to conduct data augmentation, *i.e.,* removing the adaptive augmentation module from our method. We report the results of four methods in Table 5.3. First, our method improves by on average 1.81%, compared to Proposed-R-DA method, on all data sets. This illustrates that our adaptive augmentation method is effective in preserving the intrinsic structure and significant features of the data. Second, Proposed-R-F and Proposed-R-T improve by on average 0.56% and 0.72%, respectively, compared to Proposed-R-DA, on eight data sets. This contributes to the fact that both the node feature and the topology structure are important for the UGRL. This clearly demonstrates the feasibility of the adaptive augmentation of either the node feature or the topology structure.

Table 5.5: Classification accuracy of three representations.

Datasets	Top-Feature	Fea-Feature	Pro-Feature
Citeseer	72.10±0.14	68.21± 0.33	72.87± 0.40
Cora	82.94± 0.18	81.44± 0.04	83.85± 0.35
Wiki-CS	77.36± 0.14	77.23± 0.42	79.56± 0.14
Computers	86.31± 0.25	85.24± 0.14	87.98± 0.30
Coauthors	92.33± 0.14	92.14± 0.07	93.10± 0.14
DBLP	84.21± 0.15	83.20± 0.14	85.21± 0.16
Photo	93.10± 0.11	92.18± 0.23	93.28± 0.19
Pubmed	79.25± 0.14	78.21± 0.18	80.25± 0.35

5.4.4.2 Effectiveness of contrastive losses

To validate the effectiveness of contrastive losses, we conduct experiments to compare our method with its two variants, *i.e.,* Proposed w/o n and Proposed w/o v. Specifically, Proposed w/o n denotes our method in Eq. (5.4) with only $\mathcal{L}_v$, *i.e.,* removing inter-graph contrastive loss $\mathcal{L}_n$ in Eq. (5.4). Proposed w/o v denotes our method in Eq. (5.4) with only $\mathcal{L}_n$. The results are presented in Table 5.4. The performance of our method degrades without one of them on all data sets. This demonstrates the effectiveness of our contrastive losses. Specifically, our method improves by on average 0.99% and 1.04%, respectively, compared to Proposed w/o n and Proposed w/o v, in terms of node classification. This improvement can be attributed to our multi-view contrastive learning scheme. It also indicates that it is feasible to extract the feature structure and the topology structure of the data.

5.4.4.3 Effectiveness of different graphs

We verify the effectiveness of the topology graph and the feature graph. To do this, we modify our method to output two different representations. Specifically, we first mask the inter-graph contrastive loss in our method, aiming at avoiding the feature graph to fuse information from the topology graph. We then obtain topology graph embedding $Mean(\mathbf{Z}_t, \widetilde{\mathbf{Z}}_t)$ (*i.e.,* Top-Feature) and feature graph embedding $mean(\mathbf{Z}_f, \widetilde{\mathbf{Z}}_f)$ (*i.e.,* Fea-Feature). We report the results in Table 5.5 by denoting our method as Pro-feature. Obviously, Pro-Feature achieve the best performance, followed by Top-Feature and Fea-Feature. For example, the classification accuracy of Pro-Feature improves on average by 1.75%, compared to Top-Feature, on all data sets. This illustrates that our method makes full use of the topology and feature information of the graph to generate high quality representations as they provide complementary information to each other.

5.5 Conclusion

In this chapter, we designed a novel CL-UGRL method to embed data augmentation and multi-view contrastive learning in a unified framework. To do this, this paper first proposed an adaptive data augmentation method to maintain the intrinsic structure of the data, and then designed contrastive losses to explore complementary information among two different graph structures. As a result, data augmentation and multi-view contrastive learning are iteratively adjusted to preserve the intrinsic of the data as well as extract complementary information. Experimental results on eight real data sets demonstrated the effectiveness of our method, compared to state-of-the-art comparison methods.

This chapter has been accepted in the CORE rank A* conference, *i.e.,* International Joint Conference on Artificial Intelligence.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

With the development of information industrialization, the graph data have become a common form of the data in academia and industry. Due to its large size, complex structure and multiple information, the graph data are difficult to be used effectively in industry. Therefore, the research on representation learning for the graph data has been a popular direction on both academia and industry. After analyzing existing representation learning methods for the graph data, we observe that (1) the quality of constructed graph is sensitive to either noise or outlier, and (2) single type of measurements leads to the lack of accuracy and generalization ability of similarity measurements, which hinder the improvement of graph representation learning. In light of this, this thesis proposes three novel graph representation learning methods to solve the above issues.

Chapter 3 proposed a multi-graph learning framework to conduct graph representation learning and personalized disease diagnosis on fMRI data in a semi-supervised manner. The framework investigates a multi-graph fusion method to explore both the common and the complementary information between two functional connectivity networks, and automatically learn a sparse functional connectivity network for each subject. As a result, the proposed method achieves superior performance, compared with both traditional graph learning methods and deep graph learning methods, in terms of disease diagnosis and functional neuroimaging biomarker identification.

Chapter 4 proposed a novel multi-graph fusion method to produce a high-quality graph for the GCN model. The method first extracts the common information and the complementary information among multiple local graphs to obtain a unified local graph, which is then fused with the global graph of the data to obtain the initial graph for the GCN model. Finally, the method feeds the fused graph into the GCN model to output high-quality node representation. Experimental results on real datasets demonstrated that our method outperformed the comparison methods in terms of classification tasks.

Chapter 5 proposed a multi-view unsupervised graph representation learning method. The method

embeds data augmentation with contrastive learning in a unified framework to extract common and complementary information of the topology graph and the feature graph, followed by employing the feature-level fusion method to generate new representation. As a consequence, the proposed method outperformed the state-of-the-art methods for node classification tasks and node clustering tasks.

6.2 Future work

The research of graph representation learning on large scale complex information networks is an important research topic of modern artificial intelligence. In the future work, we will focus on the following areas.

- Multiplex network representation learning. Most existing graph representation learning assume that there is only one type of relationship between nodes. However, in real applications, networks usually have multiple types of relationships. Therefore, most of the existing graph representation learning are not applicable for the study of multiplex networks. To solve this problem, it is important to develop novel multilplex network embedding methods. We will focus on information extraction and fusion of multiplex networks to develop robust multiplex representation learning methods.
- Large-scale graph representation learning. Recently, the amount of the data in various fields has been increasing, especially the emergence of large-scale social networks. Current graph representation learning methods cannot be extended to large-scale networks well. Thus, it is challenging for overcoming the issues of storage cost and efficiency in the study of real large-scale networks.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] U Rajendra Acharya, Steven Lawrence Fernandes, Joel En WeiKoh, Edward J Ciaccio, Mohd Kamil Mohd Fabell, U John Tanik, V Rajinikanth, and Chai Hong Yeong. Automated detection of alzheimer's disease using brain mri images–a study with various feature extraction techniques. *Journal of Medical Systems*, 43(9):302, 2019.

[3] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 37–48, 2013.

[4] Nesreen K Ahmed, Ryan A Rossi, Rong Zhou, John Boaz Lee, Xiangnan Kong, Theodore L Willke, and Hoda Eldardiry. Inductive representation learning in large attributed graphs. *arXiv preprint arXiv:1710.09471*, 2017.

[5] Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Dipendra Misra. Investigating the role of negatives in contrastive representation learning. *arXiv preprint arXiv:2106.09943*, 2021.

[6] Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. Fast incremental and personalized pagerank. *arXiv preprint arXiv:1006.2880*, 2010.

[7] Mukund Balasubramanian and Eric L Schwartz. The isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002.

[8] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14, 2001.

[9] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[10] Richard F Betzel and Danielle S Bassett. Multi-scale brain networks. *Neuroimage*, 160:73–83, 2017.

[11] Matthew R Brier, Anish Mitra, John E McCarthy, Beau M Ances, and Abraham Z Snyder. Partial covariance based functional connectivity computation using ledoit–wolf covariance regularization. *NeuroImage*, 121:29–38, 2015.

[12] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[13] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 891–900, 2015.

[14] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[15] Yu-Wei Chang and Jiahe Chen. What motivates customers to shop in smart shops? the impacts of smart technology and technology readiness. *Journal of Retailing and Consumer Services*, 58:102325, 2021.

[16] Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos Tsourakakis. Deepwalking backwards: from embeddings back to graphs. In *International Conference on Machine Learning*, pages 1473–1483, 2021.

[17] Feihu Che, Guohua Yang, Dawei Zhang, Jianhua Tao, and Tong Liu. Self-supervised graph representation learning via bootstrapping. *Neurocomputing*, 456:88–96, 2021.

[18] Jianhui Chen, Jieping Ye, and Qi Li. Integrating global and local structures: A least squares framework for dimensionality reduction. In *CVPR*, pages 1–8, 2007.

[19] Kejia CHEN, Zeyu YANG, Zheng LIU, and Hao LU. Graph convolutional network model using neighborhood selection strategy. *Journal of Computer Applications*, 39(12):3415, 2019.

[20] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *ICML*, pages 1725–1735, 2020.

[21] Xiaobo Chen, Han Zhang, Yue Gao, Chong-Yaw Wee, Gang Li, Dinggang Shen, and Alzheimer's Disease Neuroimaging Initiative. High-order resting-state functional connectivity network for mci classification. *Human brain mapping*, 37(9):3282–3296, 2016.

[22] Xing Chen, Yu-An Huang, Zhu-Hong You, Gui-Ying Yan, and Xue-Song Wang. A novel approach based on katz measure to predict associations of human microbiota with non-infectious diseases. *Bioinformatics*, 33(5):733–739, 2017.

[23] Yu Chen, Lingfei Wu, and Mohammed J Zaki. Deep iterative and adaptive learning for graph neural networks. *arXiv preprint arXiv:1912.07832*, 2019.

[24] Jiafeng Cheng, Qianqian Wang, Zhiqiang Tao, Deyan Xie, and Quanxue Gao. Multi-view attribute graph convolution networks for clustering. In *IJCAI*, 2020.

[25] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE transactions on knowledge and data engineering*, 31(5):833–852, 2018.

[26] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C Sinan Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 63(1):1–38, 2010.

[27] Willem de Haan, Yolande AL Pijnenburg, Rob LM Strijers, Yolande van der Made, Wiesje M van der Flier, Philip Scheltens, and Cornelis J Stam. Functional neural network analysis in frontotemporal dementia and alzheimer's disease using eeg and graph theory. *BMC neuroscience*, 10(1):101–112, 2009.

[28] Fernando De La Torre and Michael J Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1):117–142, 2003.

[29] Yao Ding, Zhili Zhang, Xiaofeng Zhao, Danfeng Hong, Wei Cai, Chengguo Yu, Nengjun Yang, and Weiwei Cai. Multi-feature fusion: Graph neural network and cnn combining for hyperspectral image classification. *Neurocomputing*, 2022.

[30] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *ICML*, pages 272–279, 2008.

[31] Harini Eavani, Theodore D Satterthwaite, Roman Filipovych, Raquel E Gur, Ruben C Gur, and Christos Davatzikos. Identifying sparse connectivity patterns in the brain using resting-state fmri. *Neuroimage*, 105:286–299, 2015.

[32] Paul Erdös. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[33] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: a library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 08 2008.

[34] Sichao Fu, Weifeng Liu, Weili Guan, Yicong Zhou, Dapeng Tao, and Changsheng Xu. Dynamic graph learning convolutional networks for semi-supervised classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(1s):1–13, 2021.

[35] Soham Gadgil, Qingyu Zhao, Adolf Pfefferbaum, Edith V Sullivan, Ehsan Adeli, and Kilian M Pohl. Spatio-temporal graph convolution for resting-state fmri analysis. In *MICCAI*, pages 528–538, 2020.

[36] Jiangzhang Gan, Rongyao Hu, Yujie Mo, Zhao Kang, Liang Peng, Yonghua Zhu, and Xiaofeng Zhu. Multigraph fusion for dynamic graph convolutional network. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[37] Jiangzhang Gan, Ziwen Peng, Xiaofeng Zhu, Rongyao Hu, Junbo Ma, and Guorong Wu. Brain functional connectivity analysis based on multi-graph fusion. *Medical image analysis*, 71:102057, 2021.

[38] Xiang Gao, Wei Hu, and Zongming Guo. Exploring structure-adaptive graph learning for robust semi-supervised classification. In *ICME*, pages 1–6, 2020.

[39] Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015.

[40] Claire M Gillan, Annemieke M Apergis-Schoute, Sharon Morein-Zamir, Gonzalo P Urcelay, Akeem Sule, Naomi A Fineberg, Barbara J Sahakian, and Trevor W Robbins. Functional neuroimaging of avoidance habits in obsessive-compulsive disorder. *American Journal of Psychiatry*, 172(3):284–293, 2015.

[41] Claire M Gillan, Martina Papmeyer, Sharon Morein-Zamir, Barbara J Sahakian, Naomi A Fineberg, Trevor W Robbins, and Sanne de Wit. Disruption in the balance between goal-directed behavior and habit learning in obsessive-compulsive disorder. *American Journal of Psychiatry*, 168(7):718–726, 2011.

[42] Vladimir Gligorijević, Yannis Panagakis, and Stefanos Zafeiriou. Non-negative matrix factorizations for multiplex network analysis. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):928–940, 2018.

[43] Geoff Gordon and Ryan Tibshirani. Karush-kuhn-tucker conditions. *Optimization*, 10(725/36):725, 2012.

[44] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[45] Michael D Greicius, Ben Krasnow, Allan L Reiss, and Vinod Menon. Functional connectivity in the resting brain: a network analysis of the default mode hypothesis. *PANS*, 100(1):253–258, 2003.

[46] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[47] Jie Gui, Zhenan Sun, Shuiwang Ji, Dacheng Tao, and Tieniu Tan. Feature selection based on structured sparsity: A comprehensive study. *IEEE transactions on neural networks and learning systems*, 28(7):1490–1507, 2016.

[48] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1024–1034, 2017.

[49] William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

[50] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, pages 4116–4126, 2020.

[51] Yuxin He, Lishuai Li, Xinting Zhu, and Kwok Leung Tsui. Multi-graph convolutional-recurrent neural network (mgc-rnn) for short-term forecasting of transit passenger flow. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[52] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[53] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[54] Rongyao Hu, Ziwen Peng, Xiaofeng Zhu, Jiangzhang Gan, Yonghua Zhu, Junbo Ma, and Guorong Wu. Multi-band brain network analysis for functional neuroimaging biomarker identification. *IEEE Transactions on Medical Imaging*, page 10.1109/TMI.2021.3099641, 2021.

[55] Huifang Huang, Xingdan Liu, Yan Jin, Seong-Whan Lee, Chong-Yaw Wee, and Dinggang Shen. Enhancing the representation of functional connectivity networks by fusing multi-view information for autism spectrum disorder diagnosis. *Human brain mapping*, 40(3):833–854, 2019.

[56] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. *Advances in neural information processing systems*, 31, 2018.

[57] Zexi Huang, Arlei Silva, and Ambuj Singh. A broader picture of random-walk based graph embedding. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 685–695, 2021.

[58] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021.

[59] Christian Janiesch, Patrick Zschech, and Kai Heinrich. Machine learning and deep learning. *Electronic Markets*, 31(3):685–695, 2021.

[60] Ziyu Jia, Youfang Lin, Jing Wang, Ronghao Zhou, Xiaojun Ning, Yuanlai He, and Yaoshuai Zhao. Graphsleepnet: Adaptive spatial-temporal graph convolutional networks for sleep stage classification. In *IJCAI*, pages 1324–1330, 2020.

[61] Bo Jiang, Xingyue Jiang, Ajian Zhou, Jin Tang, and Bin Luo. A unified multiple graph learning and convolutional network model for co-saliency estimation. In *ACMMM*, pages 1375–1382, 2019.

[62] Bo Jiang, Ziyan Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-supervised learning with graph learning-convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11313–11320, 2019.

[63] Xiaodong Jiang, Pengsheng Ji, and Sheng Li. Censnet: Convolution with edge-node switching in graph neural networks. In *IJCAI*, pages 2656–2662, 2019.

[64] Xiaodong Jiang, Ronghang Zhu, Sheng Li, and Pengsheng Ji. Co-embedding of nodes and edges with graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[65] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2651–2664, 2013.

[66] Jing Jin, Zhiqiang Wang, Ren Xu, Chang Liu, Xingyu Wang, and Andrzej Cichocki. Robust similarity measurement based on a novel time filter for ssveps detection. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[67] Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan. Multi-scale contrastive siamese networks for self-supervised graph representation learning. *arXiv preprint arXiv:2105.05682*, 2021.

[68] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *KDD*, pages 66–74, 2020.

[69] Luyang Jing, Ming Zhao, Pin Li, and Xiaoqiang Xu. A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox. *Measurement*, 111:1–10, 2017.

[70] Zhao Kang, Haiqi Pan, Steven CH Hoi, and Zenglin Xu. Robust graph learning from noisy data. *IEEE Transactions on Cybernetics*, 50(5):1833–1843, 2019.

[71] Zhao Kang, Guoxin Shi, Shudong Huang, Wenyu Chen, Xiaorong Pu, Joey Tianyi Zhou, and Zenglin Xu. Multi-graph fusion for multi-view spectral clustering. *Knowledge-Based Systems*, 189:105102, 2020.

[72] Christof Karmonik, Anthony Brandt, Saba Elias, Jennifer Townsend, Elliott Silverman, Zhaoyue Shi, and J Todd Frazier. Similarity of individual functional brain connectivity patterns formed by music listening quantified with a data-driven approach. *International journal of computer assisted radiology and surgery*, pages 1–11, 2019.

[73] Anees Kazi, Shayan Shekarforoush, S Arvind Krishna, Hendrik Burwinkel, Gerome Vivar, Benedict Wiestler, Karsten Kortüm, Seyed-Ahmad Ahmadi, Shadi Albarqouni, and Nassir Navab. Graph convolution based attention model for personalized disease prediction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 122–130, 2019.

[74] Hansang Kim, Youngbae Kim, Jae-Young Sim, and Chang-Su Kim. Spatiotemporal saliency detection for video sequences based on random walk with restart. *IEEE Transactions on Image Processing*, 24(8):2552–2564, 2015.

[75] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[76] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Machine learning proceedings 1992*, pages 249–256. 1992.

[77] Dakshina Ranjan Kisku, Phalguni Gupta, and Jamuna Kanta Sing. Feature level fusion of face and palmprint biometrics. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 512–521. Springer, 2010.

[78] Dehan Kong, Baiguo An, Jingwen Zhang, and Hongtu Zhu. L2rm: Low-rank linear regression models for high-dimensional matrix responses. *Journal of the American Statistical Association*, 115(529):403–424, 2020.

[79] Dehan Kong, Kaijie Xue, Fang Yao, and Hao H Zhang. Partially functional linear regression in high dimensions. *Biometrika*, 103(1):147–159, 2016.

[80] Byung Il Kwak, Mee Lan Han, and Huy Kang Kim. Cosine similarity based anomaly detection methodology for the can bus. *Expert Systems with Applications*, 166:114066, 2021.

[81] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

[82] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[83] Hui Li, Mengyao Zhang, and Chenbo Zeng. Circular jaccard distance based multi-solution optimization for traveling salesman problems. *Mathematical Biosciences and Engineering*, 19(5):4458–4480, 2022.

[84] Juzheng Li, Jun Zhu, and Bo Zhang. Discriminative deep random walk for network classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1004–1013, 2016.

[85] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. *arXiv preprint arXiv:1801.03226*, 2018.

[86] Zhao Li, Zhanlin Liu, Jiaming Huang, Geyu Tang, Yucong Duan, Zhiqiang Zhang, and Yifan Yang. Mv-gcn: Multi-view graph convolutional networks for link prediction. *IEEE Access*, 7:176317–176328, 2019.

[87] Zhihui Li, Feiping Nie, Xiaojun Chang, Yi Yang, Chengqi Zhang, and Nicu Sebe. Dynamic affinity graph construction for spectral clustering using multiple features. *IEEE transactions on neural networks and learning systems*, 29(12):6323–6332, 2018.

[88] Ofir Lindenbaum, Arie Yeredor, Moshe Salhov, and Amir Averbuch. Multi-view diffusion maps. *Information Fusion*, 55:127–149, 2020.

[89] Feng Liu, Wenbin Guo, Jean-Paul Fouche, Yifeng Wang, Wenqin Wang, Jurong Ding, Ling Zeng, Changjian Qiu, Qiyong Gong, Wei Zhang, et al. Multivariate classification of social anxiety disorder using whole brain functional connectivity. *Brain Structure and Function*, 220(1):101–115, 2015.

[90] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[91] Xinwang Liu, Lei Wang, Jian Zhang, Jianping Yin, and Huan Liu. Global and local structure preservation for feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6):1083–1095, 2013.

[92] Zhonghua Liu, Zhihui Lai, Weihua Ou, Kaibing Zhang, and Ruijuan Zheng. Structured optimal graph based sparse feature extraction for semi-supervised learning. *Signal Processing*, 170:107456, 2020.

[93] Minnan Luo, Xiaojun Chang, Liqiang Nie, Yi Yang, Alexander G Hauptmann, and Qinghua Zheng. An adaptive semisupervised feature analysis for video semantic recognition. *IEEE transactions on cybernetics*, 48(2):648–660, 2017.

[94] Guixiang Ma, Chun-Ta Lu, Lifang He, S Yu Philip, and Ann B Ragin. Multi-view graph embedding with hub detection for brain network analysis. In *ICDM*, pages 967–972, 2017.

[95] MD Malkauthekar. Analysis of euclidean distance and manhattan distance measure in face recognition. In *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*, pages 503–507, 2013.

[96] Tommaso Menara, Giacomo Baggio, Danielle S Bassett, and Fabio Pasqualetti. A framework to control functional connectivity in the human brain. In *CDC*, pages 4697–4704, 2019.

[97] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[98] Feiping Nie, Guohao Cai, and Xuelong Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *AAAI*, volume 31, 2017.

[99] Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. Efficient and robust feature selection via joint l2, 1-norms minimization. *Advances in neural information processing systems*, 23, 2010.

[100] Feiping Nie, Xiaoqian Wang, Michael I Jordan, and Heng Huang. The constrained laplacian rank algorithm for graph-based clustering. In *AAAI*, pages 1969–1976, 2016.

[101] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016.

[102] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114, 2016.

[103] Sandip S Patil, RP Bhavsar, and BV Pawar. Path and information content-based structural word sense disambiguation. In *International Conference on Information Processing*, pages 341–352, 2021.

[104] Liang Peng, Rongyao Hu, Fei Kong, Jiangzhang Gan, Yujie Mo, Xiaoshuang Shi, and Xiaofeng Zhu. Reverse graph learning for graph neural network. *IEEE transactions on neural networks and learning systems*, page doi.org/10.1109/TNNLS.2022.3161030.

[105] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph representation learning via graphical mutual information maximization. In *Proceedings of The Web Conference 2020*, pages 259–270, 2020.

[106] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[107] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*, pages 1150–1160, 2020.

[108] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Chi Wang, Kuansan Wang, and Jie Tang. Netsmf: Large-scale network embedding as sparse matrix factorization. In *The World Wide Web Conference*, pages 1509–1520, 2019.

[109] Nishant Rai, Ehsan Adeli, Kuan-Hui Lee, Adrien Gaidon, and Juan Carlos Niebles. Cocon: Cooperative-contrastive learning. In *CVPR*, pages 3384–3393, 2021.

[110] Zhenwen Ren and Quansen Sun. Simultaneous global and local graph structure preserving for multiple kernel clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[111] P Reyes, MP Ortega-Merchan, A Rueda, F Uriza, Hernando Santamaria-García, N Rojas-Serrano, J Rodriguez-Santos, MC Velasco-Leon, JD Rodriguez-Parra, DE Mora-Diaz, et al. Functional connectivity changes in behavioral, semantic, and nonfluent variants of frontotemporal dementia. *Behavioural neurology*, pages 1–11, 2018.

[112] Yu Rong, Tingyang Xu, Junzhou Huang, Wenbing Huang, Hong Cheng, Yao Ma, Yiqi Wang, Tyler Derr, Lingfei Wu, and Tengfei Ma. Deep graph learning: Foundations, ad-

vances and applications. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3555–3556, 2020.

[113] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[114] Christian Rubbert, Christian Mathys, Christiane Jockwitz, Christian J Hartmann, Simon B Eickhoff, Felix Hoffstaedter, Svenja Caspers, Claudia R Eickhoff, Benjamin Sigl, Nikolas A Teichert, et al. Machine-learning identifies parkinson's disease patients based on resting-state between-network functional connectivity. *The British journal of radiology*, 92(1101):20180886, 2019.

[115] Nema Salem and Sahar Hussein. Data dimensional reduction and principal components analysis. *Procedia Computer Science*, 163:292–299, 2019.

[116] Dustin Scheinost, Fuyuze Tokoglu, Michelle Hampson, Ralph Hoffman, and R Todd Constable. Data-driven analysis of functional connectivity reveals a potential auditory verbal hallucination network. *Schizophrenia bulletin*, 45(2):415–424, 2019.

[117] JM Schott, SJ Crutch, C Frost, EK Warrington, MN Rossor, and NC Fox. Neuropsychological correlates of whole brain atrophy in alzheimer's disease. *Neuropsychologia*, 46(6):1732–1737, 2008.

[118] Anil K Seth, Adam B Barrett, and Lionel Barnett. Granger causality analysis in neuroscience and neuroimaging. *Journal of Neuroscience*, 35(8):3293–3297, 2015.

[119] Niloofer Shanavas, Hui Wang, Zhiwei Lin, and Glenn Hawe. Knowledge-driven graph similarity for text classification. *International Journal of Machine Learning and Cybernetics*, 12(4):1067–1081, 2021.

[120] Chenxi Shao, Yubing Duan, and Binghong Wang. Attractive density: a new node similarity index of link prediction in complex networks. In *2015 5th International Conference on Information Science and Technology (ICIST)*, pages 74–78, 2015.

[121] Heng Tao Shen, Yonghua Zhu, Wei Zheng, and Xiaofeng Zhu. Half-quadratic minimization for unsupervised feature selection on incomplete data. *IEEE transactions on neural networks and learning systems*, page 10.1109/TNNLS.2020.3009632, 2020.

[122] Xiao Shen and Fu-Lai Chung. Deep network embedding for graph representation learning in signed networks. *IEEE transactions on cybernetics*, 50(4):1556–1568, 2018.

[123] Nicholas S Skowronski, Scott Haag, Jim Trimble, Kenneth L Clark, Michael R Gallagher, and Richard G Lathrop. Structure-level fuel load assessment in the wildland–urban interface: a fusion of airborne laser scanning and spectral remote-sensing methodologies. *International Journal of Wildland Fire*, 25(5):547–557, 2015.

[124] Sébastien Sorlin and Christine Solnon. Reactive tabu search for measuring graph similarity. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 172–182. Springer, 2005.

[125] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*, 2019.

[126] Liang Sun, Zhanhao Mo, Fuhua Yan, Liming Xia, Fei Shan, Zhongxiang Ding, Bin Song, Wanchun Gao, Wei Shao, Feng Shi, et al. Adaptive feature selection guided deep forest for covid-19 classification with chest ct. *IEEE Journal of Biomedical and Health Informatics*, 24(10):2798–2805, 2020.

[127] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Xingcheng Fu, Cheng Ji, and S Yu Philip. Graph structure learning with variational information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4165–4174, 2022.

[128] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[129] Yuuki Takai, Atsushi Miyauchi, Masahiro Ikeda, and Yuichi Yoshida. Hypergraph clustering based on pagerank. In *KDD*, pages 1970–1978, 2020.

[130] Chang Tang, Xinwang Liu, Xinzhong Zhu, Jian Xiong, Miaomiao Li, Jingyuan Xia, Xiangke Wang, and Lizhe Wang. Feature selective projection with low-rank embedding and dual laplacian regularization. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[131] Chang Tang, Xinwang Liu, Xinzhong Zhu, En Zhu, Zhigang Luo, Lizhe Wang, and Wen Gao. Cgd: Multi-view clustering via cross-view graph diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5924–5931, 2020.

[132] Chang Tang, Xiao Zheng, Xinwang Liu, Wei Zhang, Jing Zhang, Jian Xiong, and Lizhe Wang. Cross-view locality preserved diversity and consensus learning for multi-view unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[133] Jiaxiang Tang, Wei Hu, Xiang Gao, and Zongming Guo. Joint learning of graph representation and node features in graph convolutional neural networks. *arXiv preprint arXiv:1909.04931*, 2019.

[134] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826, 2009.

[135] Wei Tang, Zhengdong Lu, and Inderjit S Dhillon. Clustering with multiple graphs. In *2009 Ninth IEEE International Conference on Data Mining*, pages 1016–1021, 2009.

[136] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[137] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. *arXiv preprint arXiv:2102.06810*, 2021.

[138] Tong Tong, Katherine Gray, Qinquan Gao, Liang Chen, and Daniel Rueckert. Nonlinear graph fusion for multi-modal classification of alzheimer's disease. In *International workshop on machine learning in medical imaging*, pages 77–84. Springer, 2015.

[139] Tong Tong, Katherine Gray, Qinquan Gao, Liang Chen, Daniel Rueckert, Alzheimer's Disease Neuroimaging Initiative, et al. Multi-modal classification of alzheimer's disease using nonlinear graph fusion. *Pattern recognition*, 63:171–181, 2017.

[140] Ke Tu, Peng Cui, Xiao Wang, Fei Wang, and Wenwu Zhu. Structural deep embedding for hyper-networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[141] Nathalie Tzourio-Mazoyer, Brigitte Landeau, Dimitri Papathanassiou, Fabrice Crivello, Olivier Etard, Nicolas Delcroix, Bernard Mazoyer, and Marc Joliot. Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *Neuroimage*, 15(1):273–289, 2002.

[142] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[143] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

[144] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *ICLR (Poster)*, 2(3):4, 2019.

[145] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *KDD*, pages 1225–1234, 2016.

[146] Feng Wang, Huaping Liu, Di Guo, and Fuchun Sun. Unsupervised representation learning by invariancepropagation. *arXiv preprint arXiv:2010.11694*, 2020.

[147] Hao Wang, Yan Yang, and Bing Liu. Gmc: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1116–1129, 2019.

[148] Hua Wang, Feiping Nie, and Heng Huang. Robust distance metric learning via simultaneous l1-norm minimization and maximization. In *ICML*, pages 1836–1844, 2014.

[149] Mingliang Wang, Jiashuang Huang, Mingxia Liu, and Daoqiang Zhang. Functional connectivity network analysis with discriminative hub detection for brain disease identification. In *AAAI*, volume 33, pages 1198–1205, 2019.

[150] Tao Wang, Zexuan Ji, Jian Yang, Quansen Sun, Xiaobo Shen, Zhenwen Ren, and Qi Ge. Label group diffusion for image and image pair segmentation. *Pattern Recognition*, 112:107789, 2021.

[151] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[152] Xu Wang, Jingming He, and Lin Ma. Exploiting local and global structure for point cloud semantic segmentation with contextual point representations. In *NIPS*, pages 4571–4581, 2019.

[153] Youquan Wang, Jie Cao, and Haicheng Tao. Graph convolutional network with multi-similarity attribute matrices fusion for node classification. *Neural Computing and Applications*, pages 1–11, 2021.

[154] Zheng Wang, Feiping Nie, Lai Tian, Rong Wang, and Xuelong Li. Discriminative feature selection via a structured sparse subspace learning module. In *IJCAI*, pages 3009–3015, 2020.

[155] Chong-Yaw Wee, Pew-Thian Yap, Kevin Denny, Jeffrey N Browndyke, Guy G Potter, Kathleen A Welsh-Bohmer, Lihong Wang, and Dinggang Shen. Resting-state multi-spectrum functional connectivity networks for identification of mci patients. *PloS one*, 7(5):e37828, 2012.

[156] Chong-Yaw Wee, Pew-Thian Yap, Daoqiang Zhang, Kevin Denny, Jeffrey N Browndyke, Guy G Potter, Kathleen A Welsh-Bohmer, Lihong Wang, and Dinggang Shen. Identification of mci individuals using structural and functional connectivity networks. *Neuroimage*, 59(3):2045–2056, 2012.

[157] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2):207–244, 2009.

[158] Fei Wu, Xiao-Yuan Jing, Xinge You, Dong Yue, Ruimin Hu, and Jing-Yu Yang. Multi-view low-rank dictionary learning for image classification. *Pattern Recognition*, 50:143–154, 2016.

[159] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *ICML*, volume 97, pages 6861–6871, 2019.

[160] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153*, 2019.

[161] Hao Wu, Yu Cao, Haiping Wei, and Zhuang Tian. Face recognition based on haar like and euclidean distance. In *Journal of Physics: Conference Series*, volume 1813, page 012036, 2021.

[162] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. Unsupervised domain adaptive graph convolutional networks. In *Proceedings of The Web Conference 2020*, pages 1457–1467, 2020.

[163] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

[164] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. *Advances in Neural Information Processing Systems*, 34:28864–28876, 2021.

[165] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757*, 2021.

[166] Kai Xiong, Feiping Nie, and Junwei Han. Linear manifold regularization with adaptive graph for semi-supervised dimensionality reduction. In *IJCAI*, pages 3147–3153, 2017.

[167] Caixia Yan, Xiaojun Chang, Minnan Luo, Huan Liu, Xiaoqin Zhang, and Qinghua Zheng. Semantics-guided contrastive network for zero-shot object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[168] Liang Yang, Zesheng Kang, Xiaochun Cao, Di Jin, Bo Yang, and Yuanfang Guo. Topology optimization based graph convolutional network. In *IJCAI*, pages 4054–4061, 2019.

[169] Xi Yang, Yan Jin, Xiaobo Chen, Han Zhang, Gang Li, and Dinggang Shen. Functional connectivity network fusion with dynamic thresholding for mci diagnosis. In *MLMI*, pages 246–253, 2016.

[170] Dongren Yao, Jing Sui, Mingliang Wang, Erkun Yang, Yeerfan Jiaerken, Na Luo, Pew-Thian Yap, Mingxia Liu, and Dinggang Shen. A mutual multi-scale triplet graph convolutional network for classification of brain disorders using functional or structural connectivity. *IEEE transactions on medical imaging*, 40(4):1279–1289, 2021.

[171] Jieping Ye, Ravi Janardan, and Qi Li. Two-dimensional linear discriminant analysis. In *NIPS*, pages 1569–1576, 2005.

[172] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018.

[173] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *NIPS*, 33:5812–5823, 2020.

[174] Weiren Yu, Xuemin Lin, Wenjie Zhang, Jian Pei, and Julie A McCann. Simrank*: effective and scalable pairwise similarity search based on graph topology. *The VLDB Journal*, 28(3):401–426, 2019.

[175] Changan Yuan, Zhi Zhong, Cong Lei, Xiaofeng Zhu, and Rongyao Hu. Adaptive reverse graph learning for robust subspace learning. *Information Processing & Management*, page doi.org/10.1016/j.ipm.2021.102733, 2021.

[176] Laura Zager. *Graph similarity and matching*. PhD thesis, Massachusetts Institute of Technology, 2005.

[177] Kun Zhan, Feiping Nie, Jing Wang, and Yi Yang. Multiview consensus graph clustering. *IEEE Transactions on Image Processing*, 28(3):1261–1270, 2018.

[178] Chengyuan Zhang, Yang Wang, Lei Zhu, Jiayu Song, and Hongzhi Yin. Multi-graph heterogeneous interaction fusion for social recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(2):1–26, 2021.

[179] Han Zhang, Xiaobo Chen, Yu Zhang, and Dinggang Shen. Test-retest reliability of "high-order" functional connectivity in young healthy adults. *Frontiers in neuroscience*, 11:439, 2017.

[180] Kaihua Zhang, Tengpeng Li, Shiwen Shen, Bo Liu, Jin Chen, and Qingshan Liu. Adaptive graph convolutional network with attention graph clustering for co-saliency detection. In *CVPR*, pages 9050–9059, 2020.

[181] Li Zhang, Dan Xu, Anurag Arnab, and Philip HS Torr. Dynamic graph message passing networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3726–3735, 2020.

[182] S Zhang, Q Dong, W Zhang, H Huang, D Zhu, and T Liu. Discovering hierarchical common brain networks via multimodal deep belief network. *Medical image analysis*, 54:238–252, 2019.

[183] Shichang Zhang, Ziniu Hu, Arjun Subramonian, and Yizhou Sun. Motif-driven contrastive learning of graph representations. *arXiv preprint arXiv:2012.12533*, 2020.

[184] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: Algorithms, applications and open challenges. In *International Conference on Computational Social Networks*, pages 79–91. Springer, 2018.

[185] Tianjun Zhang, Tongzheng Ren, Mengjiao Yang, Joseph Gonzalez, Dale Schuurmans, and Bo Dai. Making linear mdps practical via contrastive representation learning. In *International Conference on Machine Learning*, pages 26447–26466, 2022.

[186] Yu Zhang, Han Zhang, Xiaobo Chen, Mingxia Liu, Xiaofeng Zhu, Seong-Whan Lee, and Dinggang Shen. Strength and similarity guided group-level brain functional network construction for mci diagnosis. *Pattern Recognition*, 88:421–430, 2019.

[187] Yu Zhang, Han Zhang, Xiaobo Chen, and Dinggang Shen. Constructing multi-frequency high-order functional connectivity network for diagnosis of mild cognitive impairment. In *IWCN*, pages 9–16, 2017.

[188] Jianyu Zhao, Zhiqiang Zhan, Qichuan Yang, Yang Zhang, Changjian Hu, Zhensheng Li, Liuxin Zhang, and Zhiqiang He. Adaptive learning of local semantic and global structure representations for text classification. In *ICCL*, pages 2033–2043, 2018.

[189] Qi Zhao, Yingjuan Yang, Guofei Ren, Erxia Ge, and Chunlong Fan. Integrating bipartite network projection and katz measure to identify novel circrna-disease associations. *IEEE transactions on nanobioscience*, 18(4):578–584, 2019.

[190] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. Robust graph representation learning via neural sparsification. In *International Conference on Machine Learning*, pages 11458–11468, 2020.

[191] Peng Zhou, Liang Du, Xuejun Li, Yi-Dong Shen, and Yuhua Qian. Unsupervised feature selection with adaptive multiple graph learning. *Pattern Recognition*, 105:107375, 2020.

[192] Runwu Zhou, Xiaojun Chang, Lei Shi, Yi-Dong Shen, Yi Yang, and Feiping Nie. Person reidentification via multi-feature fusion with adaptive graph learning. *IEEE transactions on neural networks and learning systems*, 31(5):1592–1601, 2019.

[193] Hao Zhu and Piotr Koniusz. Simple spectral graph convolution. In *International Conference on Learning Representations*, 2020.

[194] Xiaofeng Zhu, Jianye Yang, Chengyuan Zhang, and Shichao Zhang. Efficient utilization of missing data in cost-sensitive learning. *IEEE Transactions on Knowledge and Data Engineering*, page 10.1109/TKDE.2019.2956530, 2019.

[195] Xiaofeng Zhu, Shichao Zhang, Wei He, Rongyao Hu, Cong Lei, and Pengfei Zhu. One-step multi-view spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):2022–2034, 2018.

[196] Xiaofeng Zhu, Shichao Zhang, Yonghua Zhu, Pengfei Zhu, and Yue Gao. Unsupervised spectral feature selection with dynamic hyper-graph learning. *IEEE Transactions on Knowledge and Data Engineering*, page 10.1109/TKDE.2020.3017250, 2020.

[197] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

[198] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pages 2069–2080, 2021.

[199] Yonghua Zhu, Junbo Ma, Changan Yuan, and Xiaofeng Zhu. Interpretable learning based dynamic graph convolutional networks for alzheimer's disease analysis. *Information Fusion*, 77:53–61, 2022.

[200] Chenyi Zhuang and Qiang Ma. Dual graph convolutional networks for graph-based semi-supervised classification. In *Proceedings of the 2018 World Wide Web Conference*, pages 499–508, 2018.

[201] Hongliang Zou and Jian Yang. Multiple functional connectivity networks fusion for schizophrenia diagnosis. *Medical & Biological Engineering & Computing*, 2020.

Appendix A

Statement of Contribution

I confirm that the "Statement of Contribution to Doctoral Thesis Containing Publications (DRC16)", have been completed for each published article within the thesis, and are bound into the thesis and included in the electronic copy.

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Jiangzhang Gan
Name/title of Primary Supervisor:	Associate Professor Sean Zhu
Name of Research Output and full reference:	
Gan J, Peng Z, Zhu X, et al. Brain functional connectivity analysis based on multi-graph fusion[J]. Medical image analysis, 2021, 71: 102057.	
In which Chapter is the Manuscript /Published work:	Chapter 3
Please indicate:	
• The percentage of the manuscript/Published Work that was contributed by the candidate:	85
and	
• Describe the contribution that the candidate has made to the Manuscript/Published Work:	
Designing study, carrying out the experiments and results, writing manuscript, and responding the reviewers' comments.	
For manuscripts intended for publication please indicate target journal:	
N/A	
Candidate's Signature:	Jiangzhang Gan 数字签名者：Jiangzhang Gan 日期：2022.08.11 16:46:59 +12'00'
Date:	11/08/2022
Primary Supervisor's Signature:	Xiaofeng Zhu 数字签名者：Xiaofeng Zhu 日期：2022.08.26 14:22:14 +08'00'
Date:	26/08/2022

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Jiangzhang Gan
Name/title of Primary Supervisor:	Associate Professor Sean Zhu
Name of Research Output and full reference:	
Gan J, Hu R, Mo Y, et al. Multigraph Fusion for Dynamic Graph Convolutional Network[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022.	
In which Chapter is the Manuscript /Published work:	Chapter 4
Please indicate:	
• The percentage of the manuscript/Published Work that was contributed by the candidate:	85
and	
• Describe the contribution that the candidate has made to the Manuscript/Published Work:	
Designing study, carrying out the experiments and results, writing manuscript, and responding the reviewers' comments.	
For manuscripts intended for publication please indicate target journal:	
N/A	
Candidate's Signature:	Jiangzhang Gan 数字签名者：Jiangzhang Gan 日期：2022.08.11 16:46:59 +12'00'
Date:	11/08/2022
Primary Supervisor's Signature:	Xiaofeng Zhu 数字签名者：Xiaofeng Zhu 日期：2022.08.26 14:22:14 +08'00'
Date:	08/26/2022

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Jiangzhang Gan
Name/title of Primary Supervisor:	Associate Professor Sean Zhu
Name of Research Output and full reference:	
Gan J, Hu R, Zhan M, et al. Multi-view unsupervised graph representation learning[C]. The 31st International Joint Conference on Artificial Intelligence. 2022	
In which Chapter is the Manuscript /Published work:	Chapter 5
Please indicate:	
• The percentage of the manuscript/Published Work that was contributed by the candidate:	85
and	
• Describe the contribution that the candidate has made to the Manuscript/Published Work:	
Designing study, carrying out the experiments and results, writing manuscript, and responding the reviewers' comments.	
For manuscripts intended for publication please indicate target journal:	
N/A	
Candidate's Signature:	Jiangzhang Gan 数字签名者：Jiangzhang Gan 日期：2022.08.11 16:46:59 +12'00'
Date:	11/08/2022
Primary Supervisor's Signature:	Xiaofeng Zhu 数字签名者：Xiaofeng Zhu 日期：2022.08.26 14:22:14 +08'00'
Date:	26/08/2022

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)