Interpretable Graph Representation Learning: New Theories and Applications

by

Haoteng Tang

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Haoteng Tang

It was defended on

March 31st 2023

and approved by

Heng Huang, Ph.D., Professor,

Department of Electrical and Computer Engineering

Zhihong Mao, Ph.D., Professor,

Department of Electrical and Computer Engineering

Jingtong Hu, Ph.D., Associate Professor,

Department of Electrical and Computer Engineering

Bo Zeng, Ph.D., Associate Professor,

Department of Industrial Engineering

Guixiang Ma, Ph.D., Research Scientist, Intel Labs

Liang Zhan (Director), Ph.D., Associate Professor,

Department of Electrical and Computer Engineering

Interpretable Graph Representation Learning: New Theories and Applications

Haoteng Tang, PhD

University of Pittsburgh, 2023

Recent decades have witnessed the prosperity of deep learning which has revolutionized a broad variety of fields, ranging from computer vision recognition to natural language understanding. Although deep learning has achieved great success on Euclidean data (e.g., images, language sequences), the studies and explorations of the deep learning methods on graph-structured data are far from enough. The graph-structured data, presenting the relations among different items, are ubiquitous in the real world, such as transportation networks, social networks, and biological networks. However, it is challenging for regular deep learning methods to capture the hierarchical structures rooted in the graph-structured data. Another limitation of the previous graph learning models is that most of them mainly focus on unsigned graphs (i.e., graphs that only include positive and negative edges) learning. Beyond these, most of the current graph learning models are not interpretable. To address these issues, new interpretable deep graph learning models are proposed for both signed and unsigned graphs to capture the hierarchical structures in graphs and yield whole graph representations for graph-level tasks (i.e., graph classifications, and regressions). Several graph-related applications are also presented to show the practical merits of graph-structured data to the AI community.

Keywords: graph representation learning, hierarchical structure, interpretable model, signed and unsigned graphs.

Table of Contents

List of Tables

List of Figures

Preface

I would like to express my sincere gratitude to the following individuals and organizations for their support throughout the completion of this thesis.

First and foremost, I would like to thank my Ph.D. advisor, Dr. Liang Zhan, for his invaluable guidance, feedback, and encouragement throughout the research process within the past few years of my Ph.D. studies. I also would like to thank my committee members, including Dr. Heng Huang, Dr. Zhihong Mao, Dr. Jingtong Hu, Dr. Bo Zeng, and Dr. Guixiang Ma for their thoughtful feedback and insights on my research studies and my Ph.D. thesis.

I am really grateful to my research collaborators, including Dr. Paul M. Thompson, Dr. Alex Leow, Dr. Olusola Ajilore, Dr. Yalin Wang, Dr. Yong Wang, Dr. Scott Mackin, Dr. Lifang He, Dr. Emily Dennis, Dr. Haozhe Jia, Dr. Chao Li, Dr. Yong Xia, Dr. Weidong Cai, etc., for their sincerely assistance in each of my research projects and research paper in my Ph.D. studies.

I am sincerely appreciated to my lab mates, including Dr. Lei Guo, Mr. Kai Ye, Mr. Siyuan Dai, Mr. Xiyao Fu, Mr. AmirMohammad Mijani, Dr. Runxue Bao, Dr. Yanwen Wu, Mr. An Xu, Mr. Wenhan Xian, Dr. Xiaoqian Wang, Dr. Hongchang Gao, Dr. Zhouyuan Huo, Dr. Guodong Liu, Dr. Yanfu Zhang, Dr. Shangqian Gao, Mr. Junyi Li, Mr. Zhengmian Hu etc., for their assistance and encouragement in my daily works. I really enjoy my life with them during the past few years.

Additionally, I would like to thank my friends, including Mr. Qi Huang, Miss Yuxin Mei, Dr. Xingfeng Shao, Dr. Mengting Liu, Mr. Pu Jiao, Mr. Zhe Wang, Dr. Zhenwei Zhang, Dr. Shitong Mao, Miss Kechen Shu, Mr. Boyuan Yang, Mr. Zhepeng Wang, Mr. Tianqi Li, Dr. Peiyu Wang, Mr. Yike Shi, Mr. Chen Yang, Miss Yiqing Liu, Mr. Gang Qu, Miss Jiangyue Wu, Mr. Qiao Li, Mr. Shukai Ji etc., for their unwavering support, encouragement, and love throughout my academic journey, which has been instrumental in my success.

Specifically, I am deeply appreciative of my parents, Mr. Jingang Tang, and Ms Xiaoning Li, for their attentive education and firm support to my life and education.

Finally, I want to thank department of Electrical and Computer Engineering at the University of Pittsburgh, National Science Foundation (NSF), National Institutes of Health (NIH) Foundation etc., for their support that allowed me to conduct the necessary research studies for this thesis.

Thank you all for your support and encouragement.

1.0 Introduction

1.1 Background

Recent years have witnessed the prosperity of the deep learning which has revolutionized many research tasks ranging from the computer vision recognition [75,107,182] to the natural language understanding [12, 79, 138]. As a classic paradigms, the deep neural networks have shown their power on many pattern recognition and data mining tasks. For example, the deep convolution neural networks (DCNNs) are broadly used and have achieved a great success in the image classification, semantic segmentation, and object detection [32, 34, 66, 74, 85, 126, 167, 183, 227, 242]. Meanwhile, the recurrent neural networks (RNNs) also show their superiority in the tasks of speech recognition and machine translation [30, 79, 113, 184, 226]. The success of the current deep learning methods on these domains can be attributed to three aspects: (1). the collection and share of the large data resource, (2). the rapid development of the computation resources (e.g., GPUs) and (3). excellent deep learning algorithms for learning the representations from the data.

Though the deep neural networks have achieved a great success on the Euclidean data (e.g., image) and the data that can be easily encoded to the Euclidean data (e.g., text), however, the explorations and studies on the non-Euclidean data are still far from enough [214]. As a matter of fact, most of the real-world data, such as the graph data (e.g., networks), are non-Euclidean data which are used to present the relations among different items. For example, the electronic commerce network is a typical graph data which records the interactions among customers, merchants and the products. The complexity of the graph structured data introduced a few challenges in many of current machine learning methods. First, the nodes in the graph are unordered without any regular spatial structure and also the number of neighbors of graph nodes are different. This attribute of graph leads to the failure of many operations (e.g., convolution) which are broadly used in the Euclidean space data (e.g., images). Beyond of this, the basic assumption in the current machine learning tasks that the instances are independent and identically (*i.i.d.*) to each other may no longer hold in the graph data since the graph nodes are dependent by the edges among them.

In recent years, an increasing of studies [102, 150, 199] are proposed to extend the current machine learning techniques to the graph structured data. For example, motivated by the 2D convolution operation, the Graph Neural Networks (GNNs) are proposed [102] to aggregate the graph node information and embed the node features for learning node-level tasks. In general, the current GNNs can be summarized as two steps: node information aggregation and information transformation.

A critical limitation of the current GNNs is that they only propagate and aggregate the information across nodes in a 'flat' way. However, such an information propagation strategy may not capture the intrinsic structures (e.g., hierarchical structures) in the graph data [67, 127]. For many graph related tasks, especially for graph-level tasks such as graph classifications, the hierarchical structures (e.g., community structures) are critical to capture the whole graph representation that is crucial to the tasks. Therefore, part of my studies are proposed to design new graph learning methods to explore how to capture the intrinsic hierarchical structures to better represent the graph structured data.

Another limitation of the previous GNNs is that most of them mainly focus on unsigned graphs (i.e., graphs that only include positive edges) embedding but not on signed graph (i.e., graphs that include both positive and negative edges) learning. However, signed graphs, are widely utilized to depict the 'bipolar' relationships among items in this world. For example, we utilize signed graphs to describe the relationships of individuals within the social networks, where a positive edge indicates friendship between two individuals while a negative edge indicates hostility between them. Particularly, signed graphs are ubiquitous in field of brain connectomics, where functional brain networks (i.e., the brain networks derived from functional MRI) are classic signed graphs by definition. Therefore, another part of my Ph.D. studies aim to extend hierarchical graph learning models to signed graphs, where both signed graph node embedding module and hierarchical signed graph pooling module are proposed. The proposed new hierarchical signed graph learning models are mainly utilized on functional brain network embedding.

The third limitation that I focus to solve in my Ph.D. studies is that most of current graph learning models are not interpretable. Most of graph learning methods (e.g., GNNs) are black-box models, from where the results are uninterpretable and hard to understand. However, the interpretability of deep learning models is very significant especially in biomedical-related fields. A few recent studies [221] provide some interpretation for the graph neural networks, however, very few studies discuss about the interpretability in the graph pooling (or graph downscale) part. In my Ph.D. studies, a few new interpretable hierarchical graph pooling models are proposed to yield both the prediction results and result explanations as well. These models are utilized on brain network data to provide biological insights in model outcomes, such as uncovering novel biomarkers that closely related to different neurodegenerated diseases (e.g., Alzheimers' Disease) and clinical phenotypes (e.g., Human behaviors).

Beyond the theoretical studies, the graph related applications are also important to the graph mining community. Two application directions are introduced in this thesis including 1). graph reasoning for capturing long-range dependencies in semantic segmentation, and 2). graph representation learning on multimodal brain networks (e.g., functional brain networks, and structural brain networks).

1.2 Contribution

The main contributions within this thesis are summarized as follow:

- To capture the inherent hierarchical structures to represent the graph structured data, a community based hierarchical graph pooling model, named as CommPOOL, is proposed. We build up the CommPOOL to explore how to capture and utilize the community structures to obtain the whole graph representations. After we obtain the graph representations via the proposed CommPOOL, we evaluate the graph representations via the graph classification tasks.
- To embed the signed graphs in a hierarchical manner, we extend hierarchical graph pooling networks to the hierarchical signed graph pooling networks, which includes a signed graph node embedding module and a hierarchical signed graph pooling module.

- The proposed graph learning models are interpretable, which can provide the explanation of the model decisions and outcomes. Also, we visualize and discuss the model explanation explicitly.
- To graph related applications are introduced in this thesis, where we propose a pixel-wise sparse graph reasoning (PSGR) module for long-range dependencies reasoning in COVID-19 CT image segmentation, and we deploy our proposed interpretable graph learning models on brain imaging dataset (i.e., brain networks) to discover novel brain biomarkers related to neurodegenerated diseases and clinical phenotypes.

1.3 Dissertation Organization

The rest of the dissertation is organized as follows. In chapter 2, we propose an interpretable graph pooling framework, CommPOOL, for hierarchical graph representation learning. In chapter 3 and chapter 4, we extend the hierarchical graph pooling model to hierarchical signed graph pooling model for signed graph learning. Meanwhile, the applications on brain network studies are also presented within these two chapters. In chapter 5, we present our proposed graph reasoning (i.e., PSGR) module and present how it can improve the ability of DCNNs in reasoning long-range dependencies in semantic segmentation on COVID-19 CT images. The dissertation is concluded in chapter 6.

2.0 Hierarchical Graph Representation Learning

Recent years have witnessed the emergence and flourishing of hierarchical graph pooling neural networks (HGPNNs) which are effective graph representation learning approaches for graph level tasks such as graph classification. However, current HGPNNs do not take full advantage of the graph's intrinsic structures (e.g., community structure). Moreover, the pooling operations in existing HGPNNs are difficult to be interpreted. Here, we propose a new interpretable graph pooling framework — CommPOOL, that can capture and preserve the hierarchical community structure of graphs in the graph representation learning process. Specifically, the proposed community pooling mechanism in CommPOOL utilizes an unsupervised approach for capturing the inherent community structure of graphs in an interpretable manner. CommPOOL is a general and flexible framework for hierarchical graph representation learning that can further facilitate various graph-level tasks. Evaluations on five public benchmark datasets and one synthetic dataset demonstrate the superior performance of CommPOOL in graph representation learning for graph classification compared to the state-of-the-art baseline methods, and its effectiveness in capturing and preserving the community structure of graphs.

2.1 Introduction

In recent years, graph neural network (GNN) has emerged and been broadly used as a generalized deep learning architecture for graph representation learning in many fields, such as social network analysis [31, 90] and chemical molecule studies [40, 47, 65]. Generally, GNN models learn node embeddings by passing, transforming and aggregating node features across the graph. The generated node representations can then be forwarded to further layers for specific learning tasks, i.e., node classification [102, 199] and link prediction [103].

Most of the existing GNN models (e.g., GCN [102], GAT [199], GraphSage [71]) focus on node-level representation learning and only propagate information across edges of the graph in a flat way. When applying these GNNs for graph-level tasks such as graph classifications, existing works usually apply simple global pooling strategies (i.e., a summation over the learned node representations) to obtain the graph-level embedding and use it for graph label prediction [122, 201, 235]. One main drawback in these GNNs is that the hierarchical structure, often existing in graphs, is ignored during the global pooling process, which makes the models less effective for graph-level tasks. Hierarchical structure is a very important structure for many graphs in various domains. For example, the hierarchical community structure shown in Figure 1 is a typical pattern that often appears in social networks [67,127], chemical molecule networks [172] and brain networks [104, 137]. Therefore, preserving these community structures is critical for better understanding and analyzing these graphs.

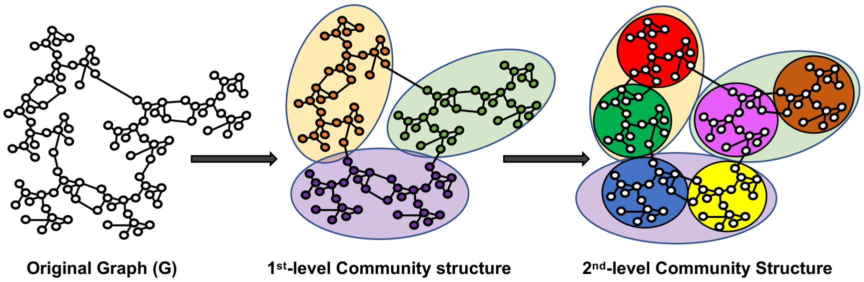


Figure 1: An example of hierarchical community structures in graphs

Some recent works proposed hierarchical graph pooling neural networks (HGPNNs) to address the hierarchical structure representation issue by introducing the hierarchical pooling operations [62,114,222,241]. Generally, these HGPNNs consist of two components: the GNN backbone which is used to embed the graph nodes and local structures, and the pooling operation which represents graph structure in a hierarchical way. These HGPNNs have demonstrated the necessity of adding hierarchical pooling operations in GNNs to better preserve the graph hierarchical structure.

However, a critical limitation of the existing hierarchical graph pooling (HGP) strategies is that few of the pooling operations in the models are interpretable. In many real applications, it is desirable to have an interpretable model, where human can understand the cause of a decision made by the model [139, 140]. Moreover, an interpretable model is more robust under adversarial attacks [41, 186, 250, 251]. A few of recent works [82, 221, 225] interpreted the node feature embedding via GNN as a neighborhood aggregation scheme. Particularly, they stated that the GNN embed the local feature of each node v within two steps: (1) neighbor node features aggregation and (2) node feature transformation. However, the interpretability of pooling operations is still not well solved (Details are discussed in the **Related Work**.). In order to make the HGP operation interpretable, three questions should be considered:

Q1: How to capture the graph hierarchical structures in an interpretable way?

Q2: How to scale down the graph representation while preserving the structures via an interpretable process?

Q3: What do we obtain after the pooling operation?

To address these challenges, we propose a Community-Based HGP framework, **COMMUNITY-POOL** or **CommPOOL**. We aim to encode the hierarchical community structure in graphs, which is a natural structure in many graphs, where nodes within each community are more densely connected than the nodes across different communities. Specifically, we propose a community-based hierarchical pooling operation which aggregates and synthesizes the node features based on the detected communities, such that the community structure of graphs can be preserved during the pooling process. Moreover, we introduce a GNN-based framework with the proposed community-based hierarchical pooling operation for learning latent graph representations, where both local node features and the hierarchical community-structure information are encoded and preserved. Our contributions can be summarized as:

- We propose a community-based HGP framework (CommPOOL) for learning graph representation in a hierarchical way that can preserve both the local node features and the hierarchical community structure of graphs.
- The proposed hierarchical community pooling strategy relies on the community structure which is explicitly detected from the graphs, therefore the pooling operation can capture the intrinsic community-level latent representation of graphs and the pooling process is inherently interpretable.
- We evaluate our CommPOOL framework for the whole graph classification task on multiple public benchmark datasets. The results demonstrate the superior performance of our model compared to several state-of-the-art graph pooling neural networks.

- Evaluations on synthetic graphs with community ground-truth labels show that our proposed CommPOOL can capture and preserve the intrinsic community structure of graphs during the learning process.

2.2 The Proposed Framework

2.2.1 Preliminaries

2.2.1.1 Graph Notation

We consider the graph classification problem on attributed graphs with different numbers of nodes. Let $G = (A, H)$ be any of the attributed graph with N nodes, where $A \in \{0,1\}^{N\times N}$ is the graph adjacency matrix and $H \in \mathcal{R}^{N\times d}$ is the node feature matrix assuming that each node has d features. Also, $Z = [Z_1, ..., Z_N]^T$ is defined as the node latent feature matrix where Z_i is the latent feature vector for the node i. Given a set of labeled data $\mathcal{D} = \{(G_1, y_1), (G_2, y_2), (G_3, y_3), ...\}$ where $y_i \in \mathcal{Y}$ is the classification label to the corresponding graph $G_i \in \mathcal{G}$. The graph classification task can be formulated as learning a mapping, f: $\mathcal{G} \rightarrow \mathcal{Y}$.

2.2.1.2 Graph Neural Network

Graph Neural Network (GNN) is an effective message-passing architecture for embedding the graph nodes and their local structures. Generally, GNN can be formulated as:

$$Z^{(k)} = F(A^{(k-1)}, Z^{(k-1)}; \theta^{(k)}), \tag{2-1}$$

where k denotes the layer k of GNN. $A^{(k-1)}$ is the graph adjacency matrix computed by layer $(k-1)$ of the GNN. $\theta^{(k)}$ is the trainable parameters in the layer k of the GNN. Particularly, $Z^0 = H$.

$F(\cdot)$ is the forward function to combine and transform the messages across the nodes. Many different versions of forward functions $F(\cdot)$ are proposed in the previous studies [65,71]

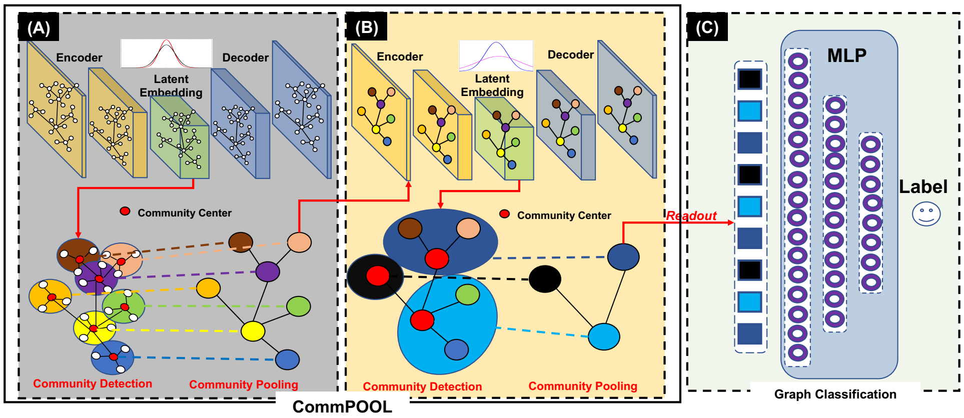


Figure 2: Framework of the CommPOOL for graph classification. **(A)** is the 1^{st} Embedding-Pooling (EP) module and **(B)** is the 2^{nd} EP module. In each module, we embed the graph into the latent space by using VGAE. In the latent space, we scale down the graph representation based on the detected communities. **(C)** is the MLP for graph classification.

such as Graph Convolutional Neural Network (GCN) [102] and Graph Attention Network (GAT) [199]. The GCN linearly combines the neighborhoods as the node the representation. And the GAT computes node representations in entire neighborhoods based on attention mechanisms [22].

2.2.2 Model Architecture

Our goal is to provide a general graph pooling framework that can capture and preserve the hierarchical community structure of graphs in the representation learning process of GNNs. The framework should be interpretable and it should be able to facilitate further graph-level learning tasks, for example, graph classification. To achieve this goal, we propose a community-based hierarchical graph pooling (HGP) framework: **CommPOOL**, which is composed of k cascaded Embedding-Pooling (EP) modules to learn the graph representation in a hierarchical way. Each EP module consists of (1) an Embedding stage, where a GNN model is employed to get the latent node representations (i.e., node embeddings) of the input graph, and (2) a Pooling stage, where a newly proposed community pooling mechanism is conducted on the node embeddings to detect communities from the graph and obtain a scaled-down graph-level representation that encodes both the local node features and the community structure of the graph. The output of the last EP module will be the final graph-level representation that preserves the overall hierarchical community structure of the graph. Figure 2(A, B) shows an instance of the proposed framework with two cascaded EP modules. In real applications of our framework, the choice of value for k is flexible and it can be decided based on the practical needs or domain knowledge for the specific application (e.g., domain evidence about how many community hierarchies exist in the graphs). In this work, we set $k = 2$ and use the architecture given in Figure 2(A, B) for illustrating our framework and we use the MLP shown in Figure 2(C) for evaluating the CommPOOL in graph representation learning for facilitating graph classification task.

In the following subsections, we introduce the two main parts in the proposed EP module for CommPOOL: (1) the GNN-based Graph Node **Embedding**, and (2) the Community **Pooling** Operation.

2.2.3 GNN-based Node Embedding

We aim at a general GNN-based model to embed the graph nodes into the latent feature space $\mathcal{Z}$ that well preserves the inherent graph structures. On the one hand, the desired node latent features should well encode the node information and the information between the node and its neighbors. On the other hand, the latent features should preserve the intrinsic structures of the graphs without task-specific influences or supervised information. On account of the above considerations, we choose the Variational Graph Auto-Encoders (VGAE) [103] for this node embedding stage. VGAE is a robust unsupervised method that can embed nodes into the latent space by reconstructing the graph itself. This can help preserve the intrinsic structures of the graph without involving any task-specific information or supervision.

2.2.3.1 Encoder

In the VGAE, we need to learn a Gaussian distribution $q(Z|H, A) = \mathcal{N}(Z|\mu, \sigma^2)$ which is used to approximate the Gaussian prior $p(Z) = \mathcal{N}(Z|0, I)$. Particularly, we utilize two GNN layers to compute the μ and σ^2 parameters of q. In the first layer, μ and σ^2 share the same GNN encoder. And in the second layer, two separate GNNs are used to generate μ and σ^2 respectively. The approximation can be achieved by maximizing the Kullback–Leibler (KL) loss between p and q:

$$\mathcal{L}_{KL} = KL(q(Z|H, A)||p(Z)) \tag{2-2}$$

The latent features Z can be obtained by resampling from the optimal $q(Z|H, A)$.

2.2.3.2 Decoder

After we obtain the latent features Z, we reconstruct the original graph adjacency matrix by:

$$\hat{A} = sigmoid(ZZ^T). \tag{2-3}$$

We define $+$ and $-$ as the edges and non-edges position index in A. So we reconstruct the adjacency matrix by minimizing the $\mathcal{L}_A$:

$$\begin{aligned} \mathcal{L}_A &= \mathcal{L}^+ + \mathcal{L}^- \\ &= -\frac{1}{E_1}\Sigma(log(\hat{A}^+)) - \frac{1}{E_2}\Sigma(log(1 - \hat{A}^-)) \end{aligned} \tag{2-4}$$

where E_1, E_2 is the number of edges and non-edges. The overall objective function of VGAE is:

$$\underset{Z \in \mathcal{Z}}{\text{minimize}} \quad \mathcal{L}_A - \mathcal{L}_{KL} \tag{2-5}$$

In our CommPOOL, we use GCN [102] to build up the basic encoder layers and use GAT [199] and Higher-Order Graph Neural Network (HO-GNN) [144] as the encoder variations.

2.2.4 Community Pooling

After embedding the graph nodes into the latent space $\mathcal{Z}$, the latent features represent the node attributes and topology structures among nodes in the latent space. The distance between two latent features measures the node attributes similarity and structure similarity between the two nodes. The proposed CommPOOL assigns the similar nodes into a community which is further used to scale down the graph.

2.2.4.1 Community Capturing

We adopt an unsupervised clustering method Partitioning Around Medoids (PAM) [97, 114] on the node latent feature vectors to group the graph nodes into L different communities, where L is a parameter denoting the number of communities in the graph. Our community partition problem can be defined as: given all the N nodes in graph G with their latent feature vectors set $V = \{Z_1, ..., Z_N\}$, find L different nodes with their latent features $Z_C = \{Z_{C_1}, ..., Z_{C_L}\} \subset V$ from the N nodes as the optimal community centers, and assign the other nodes into these L communities based on the distances between their latent feature vectors ($O = V \setminus Z_C$) and Z_C. PAM realizes the community partition problem via the following four steps.

- **Step 1.** Initialization: Randomly select L nodes with their features Z_C as the community medoid nodes.
- **Step 2.** Clustering: Compute the L_1 distances between the medoid nodes and the rest nodes based on their feature vectors, and assign each non-medoid node to its closest community; Calculate the value for the below cost function, which computes the total distance between the non-medoid node feature vectors $O_j \in O$ and their community medoid feature vectors by:

$$Cost = \Sigma_{j=1}^{N-L} |O_j - Z_{C_x}|_{L_1}, \tag{2-6}$$

  where $Z_{C_x} \in Z_C$ is the corresponding medoid of O_j.

- **Step 3.** Adjusting: Swap each medoid node by all other non-medoids and calculate the total cost for current configuration referring to Step 2; Compare the cost of current and previous configuration and keep the configuration with the smaller total cost.
- **Step 4.** Optimization: Repeat Step 2 and 3 until the configuration does not change.

2.2.4.2 Community Pooling

In order to preserve the captured community structure during the pooling process for the entire-graph representation learning, we propose a new pooling mechanism called "community pooling", which summarizes the learned node representations based on the detected community structure. Suppose $Z_{M_i} = \{Z^1_{M_i}, ..., Z^W_{M_i}\}$ is the set consisting of the latent feature vectors of all W community member nodes except for the community center nodes $Z_{C_i} \in Z_C$ in the community$-i$. Our community pooling problem can be defined as: given a community center feature $Z_{C_i} \in Z_C$, and the corresponding W community member features Z_{M_i}, compute the community representation Z_{Comm_i}. The community pooling operation computes the community$-i$'s representation by:

$$Z_{Comm_i} = Z_{C_i} + \Sigma^W_{w=1} \; Sim(Z^w_{M_i}, Z_{C_i}) Z^w_{M_i}, \tag{2-7}$$

where $Sim(\cdot)$ is a function to measure the normalized similarity ($\in$(0,1]) between each member $Z^w_{M_i}$ and the community center Z_{C_i}. In our model, we mainly define $Sim(\cdot)$ based on L_1 distance:

$$Sim(Z^w_{M_i}, Z_{C_i}) = \frac{1}{\|Z^w_{M_i} - Z_{C_i}\|_{L1}} \tag{2-8}$$

When each community representation Z_{Comm_i} is computed, we replace the center node feature Z_{Ci} by Z_{Comm_i} and remove other community member nodes. As for the graph topology structure, the preserved center nodes are connected if and only if they are connected in the original graph. To sum up, during the pooling, the community structure information and the node features are preserved onto the community center nodes. And the graph structures among the communities are presented as the topology structure of down-scaled graph with $M < N$ nodes.

Algorithm 1: Training Procedure

```
Input  : graph: $G = (A, H)$, classification label: $y$, $K$
Output: prediction: $\hat{y}$
for $k = 1, 2, ..., K$ do
    Step 1: Use $G$ to train the VGAE
    Step 2: Obtain the latent feature using trained VGAE
    Step 3: Community Pooling on latent features and generate down-scaled graph
     $G^{(k)} = (A^{(k)}, Z^{(k)}_{Comm})$. Set $G = G^{(k)}$.
end
Step 4: $Z_{graph} = GlobalReadout(Z^{(K)}_{Comm})$
Step 5: Train $MLP$ to generate $\hat{y} = MLP(Z_{graph})$
```

2.2.5 CommPOOL for Graph Classification

When the community representations $Z^{(K)}_{Comm} = [Z^{(K)}_{Comm_1}, ..., Z^{(K)}_{Comm_L}]^T$ are obtained from the last Embedding-Pooling module ($k = K$), a global readout operation is used to generate the whole graph representation Z_{graph} by averaging $Z^{(K)}_{Comm}$. Finally, an Multilayer Perceptron (MLP) utilizes Z_{graph} to make predictions for graph classification. The training procedure of CommPOOL for the graph classification task is summarized in **Algorithm 1**.

2.3 Experiment

In this section, we evaluate our CommPOOL framework using graph classification tasks. We present our experiment results in the following four subsections: (1) We introduce the dataset used in the experiments. (2) We compare the graph classification performance between CommPOOL and several competing HGPNN models. (3) We provide some variations of the CommPOOL. (4) We test our model on the simulation data to evaluate whether CommPOOL can accurately preserve the community structures in the graph.

2.3.1 Dataset

Five graph dataset are selected from the public benchmark graph data collection [99]. Table 1 summarizes the statistics of all dataset.

Table 1: Dataset Statistics: $\mathcal{V}$ and the $\mathcal{E}$ represent the nodes and edges in graph G. c represents graph classes.

Dataset	BZR	Synthie	FRANKENSTEIN	PROTEINS	AIDS
$\#\|G\|$	405	400	4337	1113	2000
Ave.$\|\mathcal{V}\|$	35.75	95.00	16.90	39.06	15.69
Ave.$\|\mathcal{E}\|$	38.36	172.39	17.88	72.82	16.20
$\#\|c\|$	2	4	2	2	2

PROTEINS and **Synthie** [17, 45, 143] are two sets of graphs representing the protein structure. The nodes are some amino acid features such as secondary structure content and amino acid propensities. Nodes are linked by edges if the amino acid is an amino acid sequence. **FRANKENSTEIN** [147] is a set of graphs representing the molecules with or without mutagenicity. The nodes represent different chemical atoms and the edges are the chemical bonds type. **BZR** [181] is a set of graphs representing the ligands for the benzodiazepine receptor. And **AIDS** [157] is set of graphs representing molecular compounds with activity against HIV or not. The molecules are converted into graphs by representing chemical atoms as nodes and the bonds as edges.

2.3.2 Graph Classification

2.3.2.1 Baseline Methods

Our baseline methods include: two graph global pooling models (**Set2Set** [201] and **SortPool** [235]), and three HGP models (**DIFFPOOL** [222], **SAGPOOL** [114] and **HGP-SL** [241]). For fair comparisons, we set two embedding-pooling modules for all HGP models including three baseline HGPs and our CommPOOL. For the baselines, we follow the original hyperparameter search strategies provided in the related papers.

2.3.2.2 Experiments Setting

Following previous works [133,222,241], we randomly split the whole dataset into training (80%) set, validation (10%) set and testing (10%) set. We repeat this randomly splitting process 10 times, and the average test performance with standard derivation is reported in Table 2. We optimize the model via Pytorch Adam optimizer. For the VGAE in the first module, the learning rate (lr) and the weight decay (wd) are searched in $\{0.0001, 0.001, 0.005, 0.01, 0.05, 0.1\}$. The dimension of two latent GNN layers are 32 and 16. For the VGAE in the second module, the lr and wd are searched in $\{0.0001, 0.001, 0.005, 0.01\}$ and the dimension of two latent GNN layers are 64 and 32. In the community pooling operation, the number of communities (L) is searched in $\{40\%, 50\%, 60\%\}$ of the number of graph nodes (N). The MLP consists of two fully connected layers with 64 and 32 neurons and a $softmax$ output layer. The lr for training the MLP is searched in $\{0.001, 0.005, 0.01\}$. We stop training if the validation loss does not decrease for 50 epochs. Our experiments are deployed on NVIDIA Tesla P100 GPUs supported by the Bridges system of Pittsburgh Supercomputing Center (PSC) [145,190]. We implement all baselines and CommPOOL using PyTorch [149] and the torch geometric library [54].

Table 2: Average graph classification test accuracy ± standard deviation (%). The best results are shown in red

Models	BZR	Synthie	FRANKENSTEIN	PROTEINS	AIDS
Set2Set	80.50 ± 1.03	22.50 ± 0.86	60.62 ± 0.27	68.08 ± 0.56	88.80 ± 0.45
SortPool	77.00 ± 1.24	32.50 ± 1.24	59.86 ± 1.22	70.11 ± 0.04	86.00 ± 2.42
DIFFPOOL	80.50 ± 1.48	57.00 ± 2.62	60.60 ± 1.62	72.43 ± 0.26	93.50 ± 1.00
SAG-POOL	82.00 ± 2.13	45.00 ± 4.21	61.73 ± 0.76	71.86 ± 0.97	93.50 ± 1.00
HGP-SL	83.00 ± 4.30	54.00 ± 0.04	59.51 ± 1.50	84.91±1.62	95.50 ± 1.00
CommPOOL	86.00±1.23	66.50±0.38	62.15±0.37	74.74 ± 0.06	98.50±0.05

2.3.2.3 Summary of Results

Table 2 summarizes the classification performances of six models on five public datasets. Our CommPOOL outperforms all baselines in the graph classification task on almost all datasets, especially on the four-class data **Synthie**. For example, our CommPOOL shows about 5.11% improvement in the classification accuracy comparing to all baselines on **BZR** data. This superiority of CommPOOL may be credited to its advanced mechanism for capturing and preserving the community structure in the pooling operation. Also, these results indicate that the community is a crucial hierarchical structure for learning the whole graph representation.

Moreover, Table 2 shows that hierarchical pooling methods generally perform better than global pooling methods, which verifies that the hierarchical pooling can better capture the graph global representations. Among all baseline models, HGP-SL relatively performs better than others, which may be attributed to the structure learning (SL) operations in the model. On PROTEINS, HGP-SL performs the best among all baseline methods and even better than ours. As discussed in [241], the structure learning in the HGP-SL is superior to preserve more topology structure of the original protein graph after pooling several times. The topology structure preserved by HGP-SL may play an important role on PROTEINS data classification, which improve its classification performance on this data. However, our proposed CommPOOL is a general approach that can capture the inherent community structure existing in different kinds of graphs, therefore, it overall outperforms the baselines across multiple datasets.

2.3.3 Ablation Studies

2.3.3.1 Model Variations

To show the flexibility of CommPool, we compare several variations of CommPOOL on PROTEINS and FRANKENSTEIN data. As noted in **The Proposed Framework** section, GAT [199] and HO-GNN [144] are used to replace GCN as VGAE encoder variations. Moreover, instead of using the reciprocal of L_1 distance, we adopt the cosine-similarity as $Sim(\cdot)$ to measure the similarity between community members Z_{M_i} and the corresponding community center Z_{C_i} in community$-i$:

$$Sim(Z^w_{M_i}, Z_{C_i}) = \frac{Z^w_{M_i} Z_{C_i}}{\|Z^w_{M_i}\| \|Z_{C_i}\|} \tag{2-9}$$

The performance of CommPool with different encoders and similarity measures are listed in Table 3, which indicates that **GAT**, compared to **GCN**, has a better performance as the encoder in CommPOOL to embed the graph nodes. In addition, Table 3 shows that L_1 distance is better than *cosine* distance when measuring the similarity between the latent features of community member nodes and the community center nodes. A possible explanation is that L_1 distance is used in the PAM clustering. Therefore, it may be better to use the same distance metric in the community partition process.

Table 3: Performance (%) of CommPOOL with different encoder settings and different similarity measures. The best results are shown in red

CommPOOL		PROTEINS	FRANKENSTEIN
GCN	L_1	74.74 ± 0.06	62.15 ± 0.37
	cosine	73.84 ± 0.13	60.18 ± 0.42
GAT	L_1	78.84 ± 0.02	63.48 ± 0.52
	cosine	76.01 ± 0.21	62.32 ± 0.39
HO-GCN	L_1	80.01 ± 0.13	64.26 ± 0.20
	cosine	77.09 ± 0.07	63.01 ± 0.11

2.3.3.2 Parameters Analysis

To investigate how the hyperparameters impact the performance of the proposed CommPOOL, we show the graph classification results achieved by our model under different number of communities (L) and different latent feature dimensions on PROTEINS and FRANKENSTEIN data. As shown in the Figure 3, the performance of models is consistent with different L values. When increasing the L value from $0.3N$ to $0.7N$, the classification results tend to incline and decline. The best results appear when $L = 0.5N$ on PROTEINS data whereas appear when $L = 0.6N$ on FRANKENSTEIN data. This may be attributed to the different intrinsic community patterns in these two datasets. We show the model performance under different latent feature dimensions in Table 4. As shown in table Table 4, four different feature dimension combinations are set as the dimensions of latent feature generated by layers of EP module 1 and EP module 2 (e.g. $[32, 16], [64, 32]$ indicates that the dimensions of

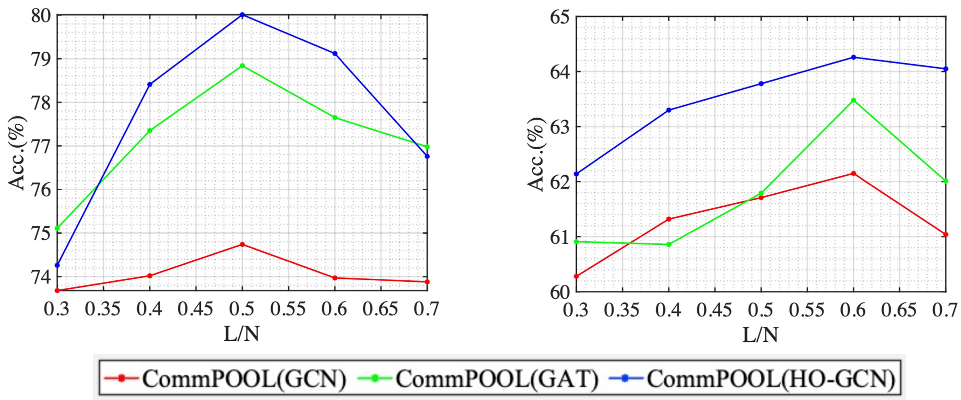


Figure 3: Graph classification accuracy (%) performed by CommPool with different VGAE encoders (e.g. GCN, GAT and HO-GCN) on (**a**) PROTEINS data and (**b**) FRANKENSTEIN data under different number of communities (L). The L values range from 0.3N to 0.7N with a step as 0.1N, where N is the number of node in each graph.

latent features generated by the two layers in the EP module 1 are 32 and 16, while the dimensions of latent features generated by the two layers in the EP module 2 are 64 and 32). It shows that, generally, the feature dimensions have slight impact on the classification results. Empirically, it indicates that lower feature dimensions should be set in EP module 1 whereas higher dimensions should be set in EP module 2.

Table 4: Average graph classification accuracy ± standard deviation (%) under different feature dimensions. The best results are shown in red.

Feature Dimensions	PROTEINS	FRANKENSTEIN
$[32, 16], [64, 32]$	74.74 ± 0.06	62.15 ± 0.37
$[64, 32], [128, 64]$	73.61 ± 0.13	62.02 ± 0.14
$[64, 32], [32, 16]$	73.89 ± 0.25	61.29 ± 1.04
$[128, 64], [64, 32]$	74.26 ± 0.31	61.90 ± 0.81

2.3.4 Community Evaluation

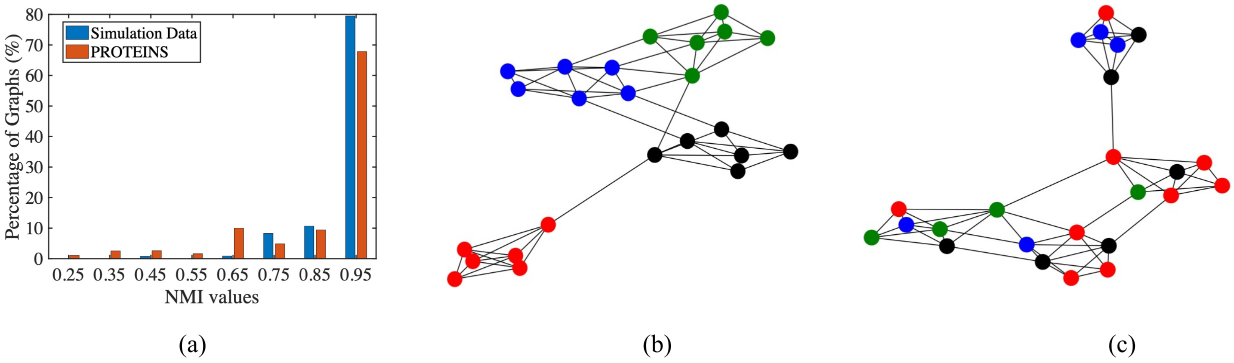
In order to evaluate if CommPool can capture the community structures, we simulate a set of graphs with the known community ground-truth and evaluate how CommPOOL preserves the intrinsic community structures on these simulation graphs. Meanwhile, on the real-data, we use PROTEINS data with the node labels as the community ground-truth to evaluate the community structures captured by the CommPOOL.

Table 5: Average graph classification accuracy ± standard deviation (%) on the simulation data. The best results are shown in red

Models	Classification Accuracy
Set2Set	46.54 ± 3.85
SortPOOL	51.29 ± 0.61
DIFF-POOL	67.14 ± 2.16
SAG-POOL	N/A
HGP-SL	72.70 ± 1.95
CommPOOL	80.14 ± 2.15

2.3.4.1 Simulation Graphs

We create 3 classes of simulation graphs using different graph generating methods, including the Random Partition Graphs, the Relax Caveman Graphs, and the Gaussian Random Community Graphs [18, 60]. Each class contains 300 graphs and each graph has 4 communities with the average size of 6 nodes. A community label is assigned to each graph node. Meanwhile, we randomly sample from the normal distribution $\mathcal{N}(0, I)$ as node features. We evaluate CommPOOL on the simulation graphs to predict their class labels. Table 5 compares the graph classification performance of CommPool with the baseline models. The results show that, on the simulation data, the CommPOOL can also outperform all the baseline models. **N/A** in Table 5 indicates the SAG-POOL cannot achieve an optimal point in reachable epochs.



(a)

(b)

(c)

Figure 4: (**a**) is the NMI histogram between the distributions of community labels of ground-truth and CommPOOL prediction. **(b)** is a positive example of community structure captured by CommPOOL leading to a correct classification. **(c)** is a negative example which leads to a graph misclassification. Different colors represent different communities.

2.3.4.2 Evaluation of Community Detection

In order to evaluate if CommPool can capture the community structures, we compare the node community label assigned by PAM clustering in the 1^{st} EP module to the community ground-truth labels. Specifically, we compute the **Normalized Mutual Information (NMI)** [176] between distribution of community labels predicted by the model and given by the ground-truth for each graph. Figure 4a is a histogram presenting the distribution of NMI scores for all 900 simulation graphs and 1113 PROTEINS data. Statistically, on the simulation data, 79.44% graphs have an NMI score larger than 0.9 and the mean NMI score is 0.9516 ± 0.098. On the PROTEINS data, 66.32% graphs have an NMI score larger than 0.9 and the mean NMI score is 0.8065 ± 0.032.

2.4 Evaluation and Discussion

In this section, we firstly discuss the interpretibility of our proposed community pooling operation. And then we analyze the importance of community structure to the graph classification task.

2.4.1 Interpretability of Community Pooling

CommPOOL is a hierarchical graph pooling framework with an interpretable pooling operation. The user can transparently understand the pooling results by monitoring the pooling operation. An interpretable pooling operation should be capable of clearly answering three questions mentioned in the **Introduction** section. Our CommPOOL provides the heuristic and knowledgeable answers for the questions in the following way:

- **Q1: How to capture the graph hierarchical structures in an interpretable way?**

  The CommPOOL considers the communities as the basic graph hierarchical structure. In the community pooling operation, we adopt PAM to group graph nodes into different communities based on the similarities among their features. The goal of the pooling operations is integrating the similar features thereby coarsening the graphs. The CommPOOL coarsens the graphs based on the graph's inherent community structure captured by the PAM, which is a natural way that allows the further pooling operation to be able to integrate the features of similar nodes. In another word, the features of nodes assigned in the same community by the CommPOOL are similar and are supposed to be integrated into one node after the pooling operation. Therefore, the overall pooling procedure is intuitive and explainable. The community capture ability of our pooling operation has been shown in the previous **Community Evaluation on Simulation Data** section.

- **Q2: How to scale down the graph while preserving the structures via an interpretable process?**

  To scale down the graph, we choose the community medoid node as the representation of the whole community, which can be understood like the centroid can be used to represent the whole mass. Meanwhile, without loss of necessary graph structure information, an interpretable structure preservation process is introduced during downscaling the graph. The community pooling achieves the preservation via gathering the nodal and structure information of the community member nodes as the features of the community medoid node.

- **Q3: What do we obtain after the pooling operation?**

  From the graph topology view, the community pooling generates community-based sub-graphs of the original graph since the pooling operation does not generate any new graph nodes and edges. Each node in the sub-graph contains the corresponding community information.

Table 6: Graph Classification Accuracy of Semi-random Pooling vs. Community-based Pooling.

Dataset	**semi-rand.**	**community-based**
PROTEINS	64.90 ± 2.45	74.74 ± 0.06
BZR	81.50 ± 2.82	86.00 ± 1.23
Synthie	59.00 ± 5.89	66.50 ± 0.38
Simulation	70.34 ± 1.26	80.14 ± 2.15

2.4.2 Community Effect on Graph Classification

We design a further experiment named ***semi-random pooling*** to show that a solid community preservation is important to the graph classification. Instead of randomly partitioning the graph into multiple communities, we only randomly select the community center nodes. After determining the community center nodes, we assign each other node to the closest community based on the similarity of node features. Such a semi-random partition method can generate a few node cliques in graphs. These cliques, though are not the optimal communities, can still maintain the hierarchical information to some degree. We replace the

PAM clustering by the semi-random partition in the pooling operation. Table 6 indicates that the community pooling has significant improvements in the graph classification tasks comparing to the semi-random pooling, which demonstrates that the success of community capture and preservation is crucial to the graph classification. To visualize, we select two simulation graphs to show (1) a positive example of community structure captured by the CommPOOL (Figure 4b); and (2) a negative example of community structure captured by the CommPOOL, which eventually leads to the graph's misclassification (Figure 4c). In addition, the performance of semi-random pooling does not decrease a lot comparing with the community pooling, which is beyond our expectations in a way. A reasonable explanation is that although unable to preserve the optimal community structure, the semi-random pooling method can still capture some degree of graph hierarchical structure, which again justifies that the significance of the community structure in the graph.

2.5 Conclusion

In this work, we propose CommPOOL, a new interpretable hierarchical graph pooling framework. CommPOOL is designed for being able to capture and preserve the inherent hierarchical community structures in graphs during the graph representation learning and scaling-down process. Moreover, CommPOOL is a general graph representation learning framework that can facilitate various graph-level tasks. Experiments on both real-world graph datasets from different domains and synthetic graph data have shown that CommPOOL outperforms the state-of-the-art methods in graph representation learning for the graph classification task.

3.0 Signed Graph Representation Learning on Multimodal Brain Networks

MRI-derived brain networks have been widely used to understand functional and structural interactions among brain regions, and factors that affect them, such as brain development and diseases. Graph mining on brain networks can facilitate the discovery of novel biomarkers for clinical phenotypes and neurodegenerative diseases. Since brain functional and structural networks describe the brain topology from different perspectives, exploring a representation that combines these cross-modality brain networks has significant clinical implications. Most current studies aim to extract a fused representation by projecting the structural network to the functional counterpart. Since the functional network is dynamic and the structural network is static, mapping a static object to a dynamic object may not be optimal. However, mapping in the opposite direction (i.e., from functional to structural networks) are suffered from the challenges introduced by negative links within signed graphs. Here, we propose a novel graph learning framework, named as Deep Signed Brain Graph Mining or DSBGM, with a signed graph encoder that, from an opposite perspective, learns the cross-modality representations by projecting the functional network to the structural counterpart. We validate our framework on clinical phenotype and neurodegenerative disease prediction tasks using two independent, publicly available datasets (HCP and OASIS). Our experimental results clearly demonstrate the advantages of our model compared to several state-of-the-art methods.

3.1 Introduction

Recent years have witnessed great progress in applying graph theory to study MRI-derived brain networks, to discover novel biomarkers for clinical phenotypes or neurodegenerative diseases (e.g., Alzheimer's disease or AD) [26,72,159,192]. Different MRI techniques can be used to reconstruct brain networks corresponding to different aspects of the brain organization or dynamics [58,153,208]. For example, functional MRI-derived functional networks provide measures of BOLD signals' relationships over time between brain regions but can not guarantee the existence of physical neuronal links among those regions [9, 56]. By contrast, diffusion MRI-derived structural network describes white matter tracts between brain regions, yet does not inform us about whether this tract, or the regions it connects, are "activated" or "not activated" in a specific state [168, 171, 185, 220]. Therefore, different types of brain networks provide distinct but complementary information, and separately analyzing each of these networks may be suboptimal.

Graph neural networks (GNNs) [102, 199] have gained enormous attentions recently. Although many progresses [15, 16, 98, 109, 174, 188, 219, 237, 238] have been made by applying GNNs to brain networks, most existing techniques are developed based on single-modality brain networks. It is well known that a high-level dependency, based on network communications, exists between brain structural and functional networks [3, 23, 24, 63, 88, 105, 128, 129, 159], which has motivated many studies [46, 55, 87, 237, 240, 248] to integrate multimodal brain networks by constructing a projection between them using GNNs (e.g., [237]). However, most existing studies aim to reconstruct functional networks from brain structural counterparts [8, 55, 87, 237]. Mapping a static object (i.e., structural network) to a dynamic one (i.e., functional network) may not be an optimal solution since the static object is better served as the template. Very few studies have been conducted for the opposite mapping (i.e., from functional to structural networks), which mainly due to that current GNNs focus on unsigned graphs (i.e., structural networks in which all edge weights are non-negative) while brain functional network is a signed graph by definition (including entries that denote negative correlations); as such, these negative network edges may undermine the mechanism of information aggregation in current GNNs [42]. Some pioneering studies (e.g., [232, 233]) convert signed functional networks to non-negative networks using absolute or threshold operations and successfully applied GNNs to embed those non-negative functional networks to reconstruct structural networks. However, those negative edges contains important information to explore the dynamics in brain functional networks and several studies have demonstrated the possibility that negative edges are related to large-scale modulation and inhibition [37, 42, 59, 68, 69, 123, 187, 228]. Therefore, these negative edges are of significant biological meanings and removing them will completely alter brain functional network's topological structure, which leads to suboptimal outcomes. We demonstrate this in the Section 3.5.5.

To tackle this, we propose a new framework to encode the signed graphs, and apply this framework to encode brain functional networks to corresponding structural networks. Our results on two publicly available data demonstrate that the extracted latent network representations from our model can be used for different clinical tasks (regression and classification) with better performances, compared with baseline methods. Our contributions are three-fold:

- We propose an end-to-end network representation learning framework to model brain structural networks from brain functional networks.
- We propose a signed graph encoder to embed the functional brain networks.
- We draw graph saliency maps for clinical tasks, to enable phenotypic and disease-related biomarker detection and aid in the biological interpretation.

3.2 Data Description and Preprocessing

Two publicly available datasets were used to evaluate our framework. The first includes data from 1206 young healthy subjects (mean age 28.19 ± 7.15, 657 women) from the Human Connectome Project [196] (HCP). The second includes 1326 subjects (mean age = 70.42 ± 8.95, 738 women) from the Open Access Series of Imaging Studies (OASIS) dataset [111]. Details of each dataset may be found on their official websites [1] [2]. Functional network was reconstructed using the standard pipeline in CONN toolbox [207]. In brief, raw EPI images were realigned, co-registered, normalized, and smoothed before analyses. Confound effects from motion artifact, white matter, and CSF were regressed out of the signal. Functional networks were then defined using Pearson Correlations of BOLD sequences between pair of ROIs. Structural network was reconstructed using FSL Probtrackx and Bedpostx functions [11]. In our study, up to three fibers were modeled per voxel. We

[1] `https://www.oasis-brains.org`

[2] `https://wiki.humanconnectome.org`

chose all voxels with $FA \geq 0.2$ as the seeds. Probtrackx was run on each individual seed voxel and repeatedly samples from the voxel-wise principal diffusion directions. This builds a distribution on the likely tract location and path, given the data. 1000 iterations were run to ensure convergence of the Markov chains, from which the posterior distributions of the local estimate of the fiber orientation distribution were sampled. For the HCP data, both networks have a dimension of 82×82 based on Destrieux atlas [44]. For the OASIS data, both networks have a dimension of 132×132 based on the Harvard-Oxford Atlas [43] and AAL Atlas [191]. We deliberately chose different network resolutions for HCP and OASIS, to evaluate whether the performance of our new framework is affected by the network dimension or atlas. Both network reconstruction procedures have been comprehensively verified in our previous studies [105, 230].

3.3 Preliminaries of Signed Brain Networks

A brain network is an attributed and weighted graph $G = \{V, E\} = (A, X)$ with N nodes, where $V = \{v_i\}_{i=1}^{N}$ is the set of graph nodes representing brain regions, and $E = \{e_{i,j}\}$ is the edge set. $X \in \mathcal{R}^{N \times d}$ is the node feature matrix where $x_i \in \mathbb{R}^{1 \times d}$ is the $i-$th row of X representing the node feature of v_i. $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix where $a_{i,j} \in \mathbb{R}$ represents the weights of the edge between v_i and v_j. In signed brain networks (i.e., brain functional networks), $a_{i,j} \in (-\infty, +\infty)$, while in unsigned brain networks (i.e., brain structural networks), $a_{i,j} \in [0, +\infty)$. We use $G^F = (A^F, X^F)$ and $G^S = (A^S, X^S)$ to represent brain functional and structural networks, respectively. Given a specific node v_i in a signed network, we define their positive and negative neighbors set as $\mathcal{N}_i^+$ and $\mathcal{N}_i^-$, respectively. Following the balance theory [29, 42, 76, 121], any node v_j belongs to the balanced set (denoted as $\mathbf{\Gamma}$) of v_i if the path between v_i and v_i contains even number of negative edges. Otherwise, v_j belongs to the unbalanced set (denoted as $\mathbf{\Upsilon}$) of v_i.

3.4 Methodology

In this section, we first introduce the proposed multi-head signed graph encoder (SGE). Then, we present an end-to-end framework with the proposed encoder to reconstruct structural networks from functional networks and perform downstream tasks.

3.4.1 Signed Graph Encoder (SGE)

Our SGE consists of a balanced-unbalanced encoder (BUE) head and a positive-negative encoder (PNE) head. Motivated by the balance theory in Section 3.3, the BUE head encodes the graph node to the latent features with balanced and unbalanced components. Meanwhile, we split the signed graph into a positive sub-graph and a negative counterpart. The PNE head encodes these two sub-graphs to generate the node latent features with positive and negative components. The yellow and blue boxes in Fig. 5 illustrate the BUE and PNE, respectively.

3.4.1.1 Balanced-Unbalanced Encoder (BUE)

We use T to denote the encoder layer number and the $T-th$ layer of the encoder focuses on aggregating the $T-th$ hop neighbors of node v_i. [3] To improve the generalization of local information aggregation in brain networks, a dynamic and fluctuating adjustment of aggregation weights (i.e., brain network edge weights) is arguably more reasonable and has been advocated during graph learning [237]. To this end, our BUE is designed based on the concept of graph attention [199].

Initial Layer $(T=1)$***.*** We initialize balanced and unbalanced node latent feature components by:

$$x_i^{\boldsymbol{\Gamma}(1)} = \sigma(\sum_{j\in\mathcal{N}_i^+} \alpha_{i,j}^{\boldsymbol{\Gamma}(1)} x_j^{(0)} W^{\boldsymbol{\Gamma}(1)}), \quad x_i^{\boldsymbol{\Upsilon}(1)} = \sigma(\sum_{j\in\mathcal{N}_i^-} \alpha_{i,j}^{\boldsymbol{\Upsilon}(1)} x_j^{(0)} W^{\boldsymbol{\Upsilon}(1)}), \tag{3-10}$$

[3] Hop is a jump describing the node connection in graph. To a target node v_i, $T-th$ hop neighbors are the nodes connected to v_i via a path with T edges.

where σ is a nonlinear activation function (i.e., $\sigma = \boldsymbol{ReLU}(\cdot)$). We use $x_i^{(0)}$ to denote the input feature of node v_i for the first layer and $x_i^{(0)}$ is initialized by the graph original node features, x_i. $W^{\mathbf{\Gamma}(1)}$ and $W^{\mathbf{\Upsilon}(1)}$ are trainable weights to generate two latent feature components. $\alpha_{i,j}^{\mathbf{\Gamma}(1)}$ and $\alpha_{i,j}^{\mathbf{\Upsilon}(1)}$ are attention scores of brain nodes from the balanced and unbalanced set respectively. Following the work in [199], we first compute the attention coefficients by:

$$e_{i,j}^{\mathbf{\Gamma}(1)} = a[x_i^{(0)}W^{\mathbf{\Gamma}(1)}, x_j^{(0)}W^{\mathbf{\Gamma}(1)}], \quad e_{i,j}^{\mathbf{\Upsilon}(1)} = a[x_i^{(0)}W^{\mathbf{\Upsilon}(1)}, x_j^{(0)}W^{\mathbf{\Upsilon}(1)}] \tag{3-11}$$

where a is a trainable attentional weight vector and $[,]$ is a concatenation operator. Based on the attention coefficients, we derive the attention scores by:

$$\alpha_{i,j}^{\mathbf{\Gamma}(1)} = \frac{exp(e_{i,j}^{\mathbf{\Gamma}(1)})}{\sum_{n\in\mathcal{N}_i^+} exp(e_{i,n}^{\mathbf{\Gamma}(1)})}, \quad \alpha_{i,j}^{\mathbf{\Upsilon}(1)} = \frac{exp(e_{i,j}^{\mathbf{\Upsilon}(1)})}{\sum_{n\in\mathcal{N}_i^-} exp(e_{i,n}^{\mathbf{\Upsilon}(1)})} \tag{3-12}$$

Following Layers (**$T > 1$).*** In the subsequent layers ($T > 1$), $x_i^{\mathbf{\Gamma}(T)}$ and $x_i^{\mathbf{\Upsilon}(T)}$ is generated by:

$$\begin{aligned} x_i^{\mathbf{\Gamma}(T)} &= \sigma(\sum_{j\in\mathcal{N}_i^+, k\in\mathcal{N}_i^-} \alpha_{i,j}^{\mathbf{\Gamma}(T)} x_j^{\mathbf{\Gamma}(T-1)} W^{\mathbf{\Gamma}(T)} + \alpha_{i,k}^{\mathbf{\Gamma}(T-1)} x_k^{\mathbf{\Upsilon}(T-1)} W^{\mathbf{\Gamma}(T)}) \\ x_i^{\mathbf{\Upsilon}(T)} &= \sigma(\sum_{j\in\mathcal{N}_i^+, k\in\mathcal{N}_i^-} \alpha_{i,j}^{\mathbf{\Upsilon}(T)} x_j^{\mathbf{\Upsilon}(T-1)} W^{\mathbf{\Upsilon}(T)} + \alpha_{i,k}^{\mathbf{\Upsilon}(T-1)} x_k^{\mathbf{\Gamma}(T-1)} W^{\mathbf{\Upsilon}(T)}) \end{aligned} \tag{3-13}$$

where $\sigma = \boldsymbol{ReLU}(\cdot)$. Similarly, we compute 4 attention coefficients as

$$\begin{aligned} e_{i,j}^{\mathbf{\Gamma}(T)} &= a[x_i^{\mathbf{\Gamma}(T-1)} W^{\mathbf{\Gamma}(T)}, x_j^{\mathbf{\Gamma}(T-1)} W^{\mathbf{\Gamma}(T)}] \\ e_{i,k}^{\mathbf{\Gamma}(T)} &= a[x_i^{\mathbf{\Upsilon}(T-1)} W^{\mathbf{\Gamma}(T)}, x_k^{\mathbf{\Upsilon}(T-1)} W^{\mathbf{\Gamma}(T)}] \\ e_{i,j}^{\mathbf{\Upsilon}(T)} &= a[x_i^{\mathbf{\Upsilon}(T-1)} W^{\mathbf{\Upsilon}(T)}, x_j^{\mathbf{\Upsilon}(T-1)} W^{\mathbf{\Upsilon}(T)}] \\ e_{i,k}^{\mathbf{\Upsilon}(T)} &= a[x_i^{\mathbf{\Gamma}(T-1)} W^{\mathbf{\Upsilon}(T)}, x_k^{\mathbf{\Gamma}(T-1)} W^{\mathbf{\Upsilon}(T)}] \end{aligned} \tag{3-14}$$

Based on these 4 attention coefficients, we compute 4 attention scores by:

$$\begin{aligned} \alpha_{i,j}^{\mathbf{\Gamma}(T)} = \frac{exp(e_{i,j}^{\mathbf{\Gamma}(T)})}{\sum_{\mathbf{\Gamma}(T)}}, \quad \alpha_{i,k}^{\mathbf{\Gamma}(T)} = \frac{exp(e_{i,k}^{\mathbf{\Gamma}(T)})}{\sum_{\mathbf{\Gamma}(T)}}, \\ \alpha_{i,j}^{\mathbf{\Upsilon}(T)} = \frac{exp(e_{i,j}^{\mathbf{\Upsilon}(T)})}{\sum_{\mathbf{\Upsilon}(T)}}, \quad \alpha_{i,k}^{\mathbf{\Upsilon}(T)} = \frac{exp(e_{i,k}^{\mathbf{\Upsilon}(T)})}{\sum_{\mathbf{\Upsilon}(T)}} \end{aligned} \tag{3-15}$$

After we obtain the two latent components, we concatenate them as the latent features generated by the BUE head by: $x_i^{BUE} = [x_i^{\mathbf{\Gamma}}, x_i^{\mathbf{\Upsilon}}]$.

3.4.1.2 Positive-Negative Encoder (PNE)

We split the adjacency matrix of the functional network into positive and negative sub-network pairs (i.e., $G^{+F} = (A^{+F}, X)$ and $G^{-F} = (A^{-F}, X)$). For each sub-network, we forward it into T graph attention layers [199] to generate the positive and negative latent feature components (i.e., x_i^+ and x_i^-). As a result, the latent feature generated by the PNE head can be computed by $x_i^{PNE} = [x_i^+, x_i^-]$. The final fused latent features generated by our SGE can be computed by: $x_i = [x_i^{BUE} W, x_i^{PNE} W]$, where W is a set of trainable parameters to control feature dimensions.

3.4.2 Deep Signed Brain Networks Model (DSBGM)

Our DSBGM framework is illustrated in Fig. 5, which includes (1) a signed graph encoder (SGE) to generate the latent node features from brain functional networks, (2) a decoder to reconstruct brain structural networks from latent node features, and (3) Multilayer Perceptrons (MLP) for downstream tasks.

3.4.2.1 Structural Network Reconstruction and Downstream Tasks

The structural network edges can be reconstructed by an inner-product decoder [103] as: $\hat{a}_{i,j}^S = sigmoid(x_i^\top \cdot x_j)$, where the $\cdot$ is an inner-product operator, $\top$ is vector transpose.

We use a sum function as a global readout operator to obtain the whole graph representation (i.e., $X_G = \sum_{i=1}^N x_i$). Then, MLP are used to generate the final classification or regression output (i.e., $\hat{y} = MLP(X_G)$). Moreover, we use parameters in the last MLP layer and node latent features x_i to generate the brain network saliency map using the Class Activation Mapping (CAM) approach [6, 151] for the classification task. Particularly, assume that the fused latent feature x_i generated by SGE is in the dimension of $\mathcal{R}^{1\times D}$, and the parameters in the last MLP layer (i.e., W_{MLP}) are in the dimension of $\mathcal{R}^{D\times C}$ where C is the output dimension. The saliency value for node v_i with respect to each class is computed by: $x_i \cdot W_{MLP}$ which is in the dimension of $\mathcal{R}^{1\times C}$. We compute a saliency value for each brain node (i.e., v_i for $i \in \{1, 2, ..., N\}$) with respect to each class to generate a saliency map for the whole brain network. The generated saliency map highlights important brain regions based on saliency values for each class.

3.4.2.2 Loss Functions

We summarize the loss functions here for our framework.

Reconstruction Loss. The ground-truth brain structural network is sparse while the reconstructed structural network is fully connected. To facilitate network reconstruction, in the training stage, we add a small perturbation value (δ) to the edge weights of the ground-truth structural networks (i.e., $\bar{a}^S_{i,j} = a^S_{i,j} + \delta$) and build up the reconstruction loss as $\mathcal{L}_{recon} = \frac{1}{|E|} \sum_{i,j} (\hat{a}^S_{i,j} - \bar{a}^S_{i,j})^2$, where $|E|$ is the number of edges.

Supervised Loss. We deploy our framework on both regression and classification tasks. For the classification task, we use the negative log likelihood loss where $\mathcal{L}_{super} = NLLLoss(\hat{y}, y)$. For the regression task, we use L_1 loss where $\mathcal{L}_{super} = L_1 Loss(\hat{y}, y)$. In summary, the overall loss function of the DSBGM is $\mathcal{L}_{all} = \eta_1 \mathcal{L}_{recon} + \eta_2 \mathcal{L}_{super}$, where η_1, η_2 are loss weights.

3.5 Experiments

3.5.1 Implementation Details

The edge weights of the functional networks and structural networks were first normalized to the range of $[-1, 1]$ and $[0, 1]$, respectively. The node features are initialized as the min, 25%, median, 75%, max values of the mean fMRI bold signal in that node (or brain region). We randomly split each dataset into 5 disjoint sets for 5-fold cross-validations in the following experiments. The model is trained using the Adam optimizer with a batch size of 128. The initial learning rate is set to 0.001 and decayed by $(1 - \frac{current\, epoch}{max\, epoch})^{0.9}$. We also regularize the training with an L_2 weight decay of $1e^{-5}$. We stop the training if the validation loss does not improve for 100 epochs in an epoch termination condition with a maximum of 500 epochs, as was done in [115, 165]. The experiments are deployed on NVIDIA TITAN RTX GPU.

3.5.2 Brain Structural Network Reconstruction using DSBGM

To show the performance of our DSBGM on the reconstruction of brain structural networks, we train the model in a task-free manner where no task-specific supervised loss is involved. Since the δ value is set to 0.05 when we train the model, we set the reconstructed edge weights to 0 if the predicted weights are less than 0.05. The mean absolute error (MAE) values between the edge weights in the ground-truth and reconstructed networks are 0.074 ± 0.016 and 0.039 ± 0.058 under 5-fold cross-validation on the OASIS and HCP data, respectively. The reconstruction results on OASIS data are visualized in Fig. 6 and the correlation between the predicted structural network and ground truth is 0.917 with p value = 0.0129.

3.5.3 Disease and Sex Classification Tasks

Experimental Setup. 7 baseline methods were used for comparison. The baseline methods include 2 traditional graph embedding models (i.e., t-BNE [27] and MK-SVM [48]), 2 deep graph convolution models designed for brain network embedding (i.e., BrainChey [109] and BrainNet-CNN [98]) and 2 hierarchical graph neural networks with graph pooling strategies (i.e., DIFFPOOL [222] and SAGPOOL [115]). We also introduce variational graph auto-encoders (VGAE) [103] with GCN as graph encoders to our baseline methods to integrate multimodal brain networks by reconstructing the structural networks from functional networks.

As mentioned above, baseline methods can only embed unsigned graphs, we therefore adopt two different strategies to convert signed functional networks to unsigned networks as the input of baseline methods. On the one hand, we compute the absolute values of the functional network edge weights; on the other hand, we only preserve positive edges and drop negative ones to yield unsigned networks. 3 variant models of our DSBGM, including DSBGM w/o BUE encoder, DSBGM w/o PNE encoder and DSBGM w/o reconstruction decoder, are evaluated for ablation studies. The number of BUE and PNE encoder layers is set to 3. We search the loss weights (see details in Fig. 7) η_1 and η_2 in the range of $[0.01, 0.1, 0.5, 1]$ and $[0.1, 1, 5]$ respectively, and determine the loss weights as $\eta_1 = 0.1$,

$\eta_2 = 1$ for AD classification, $\eta_1 = 0.1$, $\eta_2 = 1$ for sex classification. The results are reported in terms of classification accuracy, precision and F1-scores, with their *std*.

Results. Classification results for AD on OASIS and sex on HCP are summarized in Table 7a and 7b, where Table 7a presents the results with the absolute edge weights as inputs of baseline methods and Table 7b presents the results with the positive-only edges as inputs of baseline methods. These results demonstrate that our model achieves the best accuracy, comparing with all baseline methods using two different input settings, on both classification tasks. For example, in the AD classification, our model outperforms baselines with at least 7.6%, 7.8% and 9.4% increases in accuracy, precision and F1 scores respectively when using absolute edge weights as inputs of baseline methods. And it outperforms baselines with at least 8.47%, 11.90% and 11.08% increases in accuracy, precision and F1 scores respectively when using positive-only edges as inputs of baseline methods. Either computing absolute edge weights or dropping negative edges leads to an information deficiency of signed functional networks, which may partially account for performance decreases on baseline methods. Moreover, results from baseline methods with two input strategies are not consistent. For example, using absolute edge weights outperforms using only-positive edges for DIFFPOOL on gender classification, however, using only-positive edges is better for SAGPOOL on the same task. Therefore, no recommendation can be made between these two input strategies for baseline methods. In general, deep graph models perform better than traditional graph embedding methods (i.e., t-BNE and MK-SVM). When we abandon the cross-modality learning (i.e., DSBGM w/o Recon.), the performance, though comparable to baselines, decreases significantly. This shows the effectiveness of the cross-modality learning. The performance will also decrease when we remove BUE encoder or PNE encoder, which indicates that both balance and polarity properties are essential for embedding signed graphs (e.g., functional brain networks). Although VGAE and our DSBGM integrate multimodal brain networks in a reconstruction manner, our model takes signed functional networks as the input while VGAE can only take unsigned ones as the input. Our model outperforms VGAE in both classification tasks. This suggests that negative edges contain important information for network's topological structure, which is essential for network representation learning. Ignoring or removing these negative edges may lead to significant information loss. The same conclusion is also demonstrated in regression tasks in next section.

The brain saliency map is shown in Fig. 8 where 10 key brain regions associated with AD (from OASIS data) and with each sex (from HCP data) are identified, respectively. The salient regions for females are concentrated in frontal regions of the brain while males have a different trend, consistent with the finding in [28] that women are typically less aggressive than men, and, on average, less physically strong. For AD, most of the salient regions are located in subcortical structures, as well as the bilateral intracalcarine region, the caudate, and planum polare, which have been implicated as potential AD biomarkers in the literature [4]. The name of these brain regions are summarized in Table 9

3.5.4 MMSE Regression

Experimental Setup. Mini-Mental State Exam (MMSE) is a quantitative measure of cognitive status in adults. In the MMSE regression task, the selected baselines for comparison (except for MK-SVM which is designed only for classification problems) and the structure of our DSBGM remain unchanged. The loss weights are set to $\eta_1 = 0.5$ and $\eta_2 = 1$ for both datasets. Details of hyperparameter analysis are provided in the Fig. 7. The regression results are reported as average Mean Absolute Errors (MAE) with their *std*.

Results. MMSE regression results on the HCP and OASIS data are summarized in Table 8a and 8b, where Table 8a presents the results with the absolute edge weights as inputs for baseline methods and Table 8b presents the results with the positive-only edges as inputs for baseline methods. Our results show that the DSBGM outperforms all baselines using any of two input settings. And this clearly demonstrates the same three conclusions as those in classification tasks in the above section, which are (1) the superiority of the cross-modality learning; (2) the essential of negative edges in the signed graph (i.e., brain functional network); and (3) the importance of both BUE and PNE encoders. Moreover, our results indicate that the regression performance on HCP dataset is better than on OASIS dataset, which may be attributed to the range of original data (i.e., data *std*). The original mean ± *std* of MMSE are 28.38 ± 2.84 and 28.99 ± 2.26 for OASIS and HCP, respectively. And a lower deviation value of original data may result in a more precise regression result.

3.5.5 Group Analysis on B-U set and P-N set

To further demonstrate that the balanced-unbalanced (B-U) node sets and positive-negative (P-N) node sets are important signed graph topological structures, we compare these two node sets between disease groups (i.e., AD vs. NC groups) on OASIS dataset. Specifically, we search B-U sets and P-N sets on functional brain networks and compute two ratios (r_{B-U} and r_{P-N}) for each subject.

$$r_{B-U} = \frac{\#\ of\ B\ sets}{\#\ of\ U\ sets}, \quad r_{P-N} = \frac{\#\ of\ P\ sets}{\#\ of\ N\ sets}. \tag{3-16}$$

r_{B-U} value will change with the order (i.e., 1^{st}, 2^{nd} and 3^{rd} order) so that each subject will have multiple r_{B-U} values. While for P-N sets, only the 1^{st} order is meaningful and there is no higher order for P-N sets, so there is only one r_{P-N} value for each subject.

We conduct student t test on these r_{B-U} and r_{P-N} values between AD and NC groups. The p values of B-U sets in 1^{st}, 2^{nd} and 3^{rd} order are 3.2×10^{-4}, 6.9×10^{-5} and 2.7×10^{-5}, respectively. The p value of the 1^{st} order P-N set is 3.2×10^{-4}. These results suggest that there exist significant groups differences in both balanced-unbalanced node sets and positive-negative node sets, and these node sets are important topological features to distinguish the patterns of functional networks.

3.6 Conclusion

We propose a novel multimodal brain network representation learning framework with a signed functional network encoder. The cross-modality network embedding is generated by mapping a functional brain network to its structural counterpart. Our results indicate negative edges are essential for graph topological representation learning, which is consistent with the literature [228]. Moreover, we showed that BUE and PNE encoders are both necessary for the signed graph learning. The embedded network representations contribute to important clinical prediction tasks and the brain saliency map may assist with disease-related biomarker identification. We will explore the bijection mapping between these two brain networks in the future.

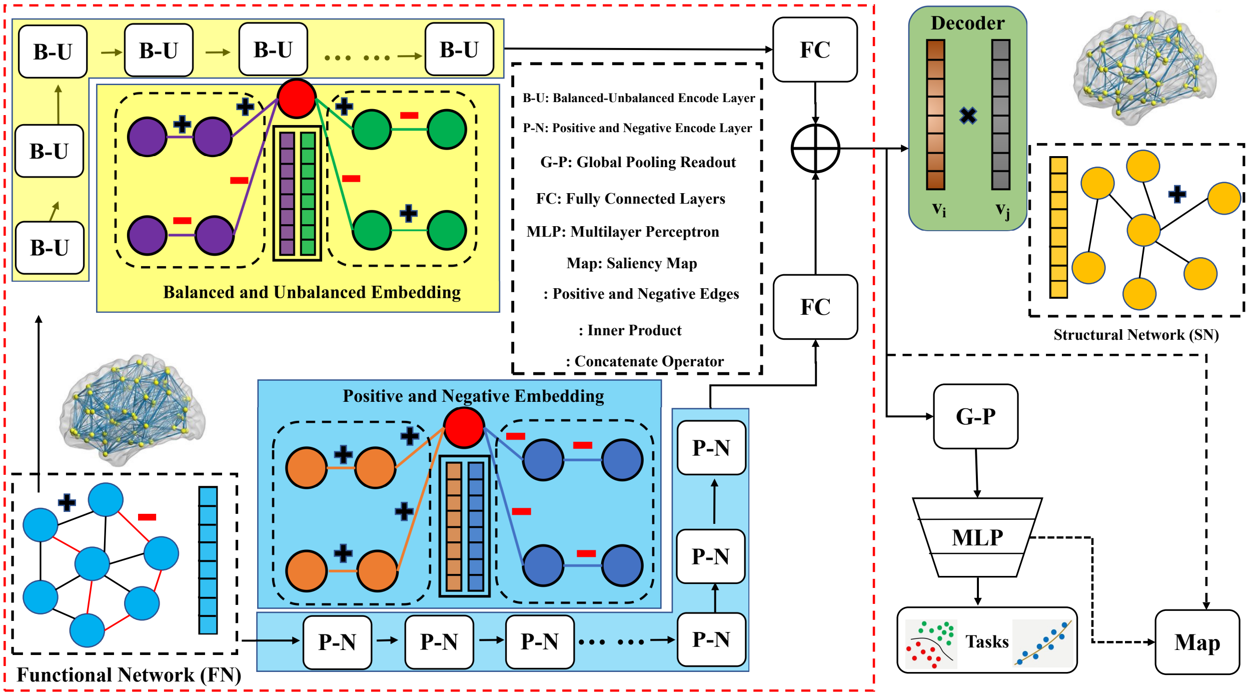

B-U
B-U
B-U
B-U
B-U
B-U
FC
Decoder
v_i
v_j
B-U: Balanced-Unbalanced Encode Layer
P-N: Positive and Negative Encode Layer
G-P: Global Pooling Readout
FC: Fully Connected Layers
MLP: Multilayer Perceptron
Map: Saliency Map
: Positive and Negative Edges
: Inner Product
: Concatenate Operator
Balanced and Unbalanced Embedding
FC
Structural Network (SN)
Positive and Negative Embedding
G-P
P-N
P-N
MLP
Tasks
Map
Functional Network (FN)
P-N
P-N
P-N
P-N

Figure 5: Pipeline of the DSBGM framework, including a signed graph encoder (in red dash blox) scheme with two branches (BUE and PNE encoder heads in yellow and blue box) for functional network embedding, an inner-product decoder for structural network reconstruction and a downstream task branch for classification and regression.

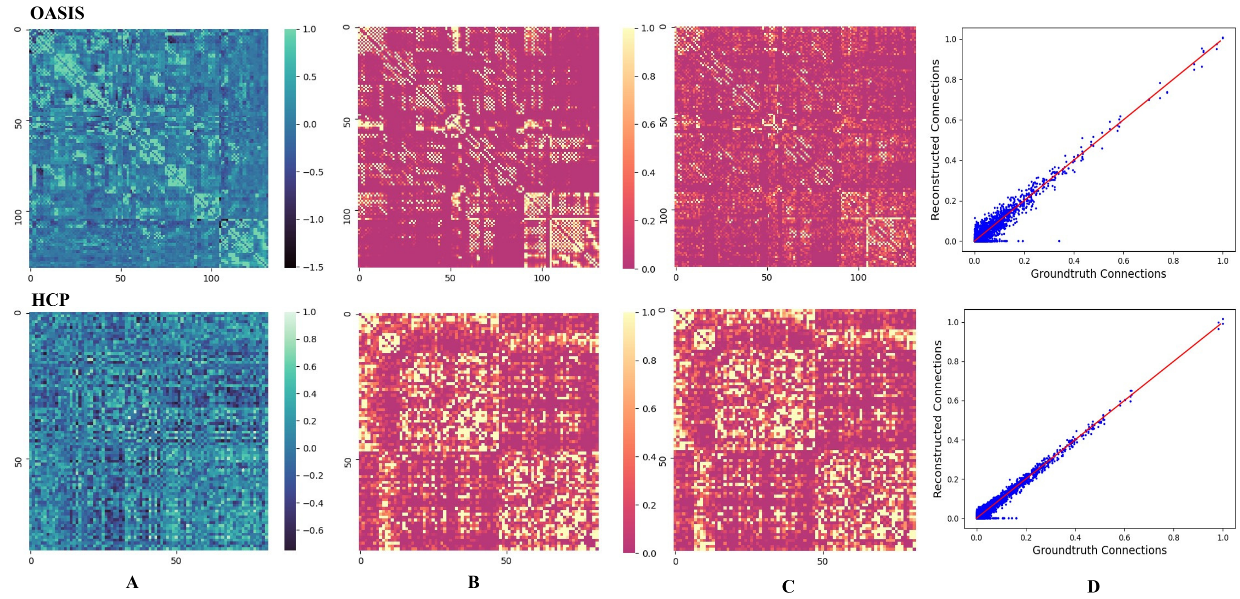


Figure 6: Cross-modality learning results on the OASIS and HCP data. **(A)** are the averaged functional networks, **(B)** and **(C)** are the mean reconstructed and ground-truth structural networks. **(D)** demonstrate that edge weights in the predicted structural network are significantly correlated with the ground truth data (r = 0.917 with p = 0.0129 for OASIS data, and r = 0.945 with p = 0.0086 for HCP data.)

Table 7a: Classification accuracy, precision, and F1-scores, with their standard deviation values under 5-fold cross-validation. Signed functional networks are converted to unsigned networks as inputs of all baseline methods by computing the absolute values of the edge weights. The best results are highlighted in **bold** font.

Method	OASIS (Disease)			HCP (Gender)		
	Acc.	Prec.	F1	Acc.	Prec.	F1
t-BNE	57.84 ± 2.14	52.26 ± 2.39	55.17 ± 3.15	59.84 ± 2.05	57.77 ± 2.21	55.89 ± 1.96
MK-SVM	54.62 ± 3.33	49.86 ± 3.11	55.27 ± 4.15	58.21 ± 3.75	51.51 ± 2.84	59.19 ± 2.40
DIFFPOOL	66.29 ± 2.60	63.87 ± 2.14	69.91 ± 2.37	72.12 ± 1.81	68.84 ± 1.97	74.01 ± 2.16
SAGPOOL	64.53 ± 2.06	61.76 ± 2.99	68.97 ± 3.07	69.29 ± 1.84	70.18 ± 1.49	67.38 ± 1.12
BrainChey	69.26 ± 1.47	71.22 ± 1.95	70.46 ± 2.81	74.11 ± 2.05	76.26 ± 1.77	75.08 ± 2.60
BrainNet-CNN	73.37 ± 2.14	74.06 ± 1.59	73.27 ± 1.85	71.88 ± 1.69	70.18 ± 2.21	70.29 ± 2.10
VGAE	72.11 ± 2.66	70.08 ± 1.30	71.55 ± 0.96	73.59 ± 2.44	73.43 ± 1.80	72.77 ± 1.03
DSBGM w/o BUE	73.04 ± 3.03	74.74 ± 1.96	73.53 ± 2.23	78.15 ± 2.96	79.01 ± 1.68	80.47 ± 2.02
DSBGM w/o PNE	75.38 ± 2.52	76.26 ± 2.96	78.68 ± 3.02	80.78 ± 2.44	81.16 ± 1.74	82.98 ± 2.01
DSBGM w/o Recon.	75.94 ± 2.73	77.09 ± 2.52	75.46 ± 2.17	79.12 ± 2.06	80.55 ± 2.18	81.19 ± 1.96
DSBGM	**78.92 ± 1.38**	**79.81 ± 1.41**	**80.22 ± 2.25**	**82.19 ± 2.01**	**85.35 ± 1.99**	**84.71 ± 2.37**

Table 7b: Classification accuracy, precision, and F1-scores, with their standard deviation values under 5-fold cross-validation. Signed functional networks are converted to unsigned networks as inputs of all baseline methods by dropping all negative edges. The best results are highlighted in **bold** font.

Method	OASIS (Disease)			HCP (Gender)		
	Acc.	Prec.	F1	Acc.	Prec.	F1
t-BNE	56.26 ± 2.98	56.03 ± 2.48	57.71 ± 2.41	61.26 ± 1.91	60.17 ± 2.49	62.29 ± 3.57
MK-SVM	55.29 ± 3.02	50.34 ± 2.83	57.27 ± 1.73	57.97 ± 3.78	53.12 ± 2.31	58.82 ± 2.82
DIFFPOOL	68.97 ± 1.34	66.03 ± 3.36	69.24 ± 1.83	67.77 ± 3.56	65.25 ± 2.65	68.82 ± 1.72
SAGPOOL	65.65 ± 2.01	63.33 ± 1.95	67.27 ± 2.09	70.95 ± 2.88	69.83 ± 1.85	71.44 ± 1.29
BrainChey	72.76 ± 2.20	71.27 ± 2.58	72.21 ± 3.55	75.01 ± 1.82	75.70 ± 1.51	74.22 ± 2.72
BrainNet-CNN	69.54 ± 2.97	70.03 ± 1.24	68.22 ± 1.40	70.92 ± 1.79	70.41 ± 2.19	71.11 ± 1.71
VGAE	64.68 ± 2.49	62.57 ± 2.19	65.85 ± 1.91	73.59 ± 2.42	74.43 ± 1.84	76.25 ± 1.49
DSBGM w/o BUE	73.04 ± 3.03	74.74 ± 1.96	73.53 ± 2.23	78.15 ± 2.96	79.01 ± 1.68	80.47 ± 2.02
DSBGM w/o PNE	75.38 ± 2.52	76.26 ± 2.96	78.68 ± 3.02	80.78 ± 2.44	81.16 ± 1.74	82.98 ± 2.01
DSBGM w/o Recon.	75.94 ± 2.73	77.09 ± 2.52	75.46 ± 2.17	79.12 ± 2.06	80.55 ± 2.18	81.19 ± 1.96
DSBGM	**78.92 ± 1.38**	**79.81 ± 1.41**	**80.22 ± 2.25**	**82.19 ± 2.01**	**85.35 ± 1.99**	**84.71 ± 2.37**

Table 8a: Regression Mean Absolute Error (MAE) ± *std* under 5-fold cross-validation. Signed functional networks are converted to unsigned networks as the input of all baseline methods by computing the absolute values of the edge weights. Lower MAE values indicate better results.

The best results are highlighted in **bold** font.

	OASIS (MMSE) ↓	HCP (MMSE) ↓
tBNE	2.39 ± 0.74	2.17 ± 0.48
SAGPOOL	1.86 ± 0.27	1.55 ± 0.33
DIFFPOOL	1.69 ± 0.36	1.63 ± 0.14
BrainNet-CNN	1.40 ± 0.20	1.29 ± 0.06
BrainChey	1.12 ± 0.19	1.14 ± 0.25
VGAE	1.27 ± 0.25	1.31 ± 0.20
DSBGM w/o BUE	1.17 ± 0.22	1.11 ± 0.17
DSBGM w/o PNE	1.06 ± 0.24	0.86 ± 0.34
DSBGM w/o Recon.	0.97 ± 0.11	1.03 ± 0.19
DSBGM	**0.87 ± 0.18**	**0.69 ± 0.21**

Table 8b: Regression Mean Absolute Error (MAE) ± *std* under 5-fold cross-validation. Signed functional networks are converted to unsigned networks as the input of all baseline methods by dropping all negative edges. Lower MAE values indicate better results.

The best results are highlighted in **bold** font.

	OASIS (MMSE) ↓	HCP (MMSE) ↓
tBNE	2.51 ± 0.21	2.02 ± 0.93
SAGPOOL	1.73 ± 0.79	1.46 ± 0.29
DIFFPOOL	1.77 ± 0.56	1.39 ± 0.15
BrainNet-CNN	1.39 ± 0.42	1.44 ± 0.09
BrainChey	1.19 ± 0.67	1.06 ± 0.49
VGAE	1.11 ± 0.16	1.21 ± 0.28
DSBGM w/o BUE	1.17 ± 0.22	1.11 ± 0.17
DSBGM w/o PNE	1.06 ± 0.24	0.86 ± 0.34
DSBGM w/o Recon.	0.97 ± 0.11	1.03 ± 0.19
DSBGM	**0.87 ± 0.18**	**0.69 ± 0.21**

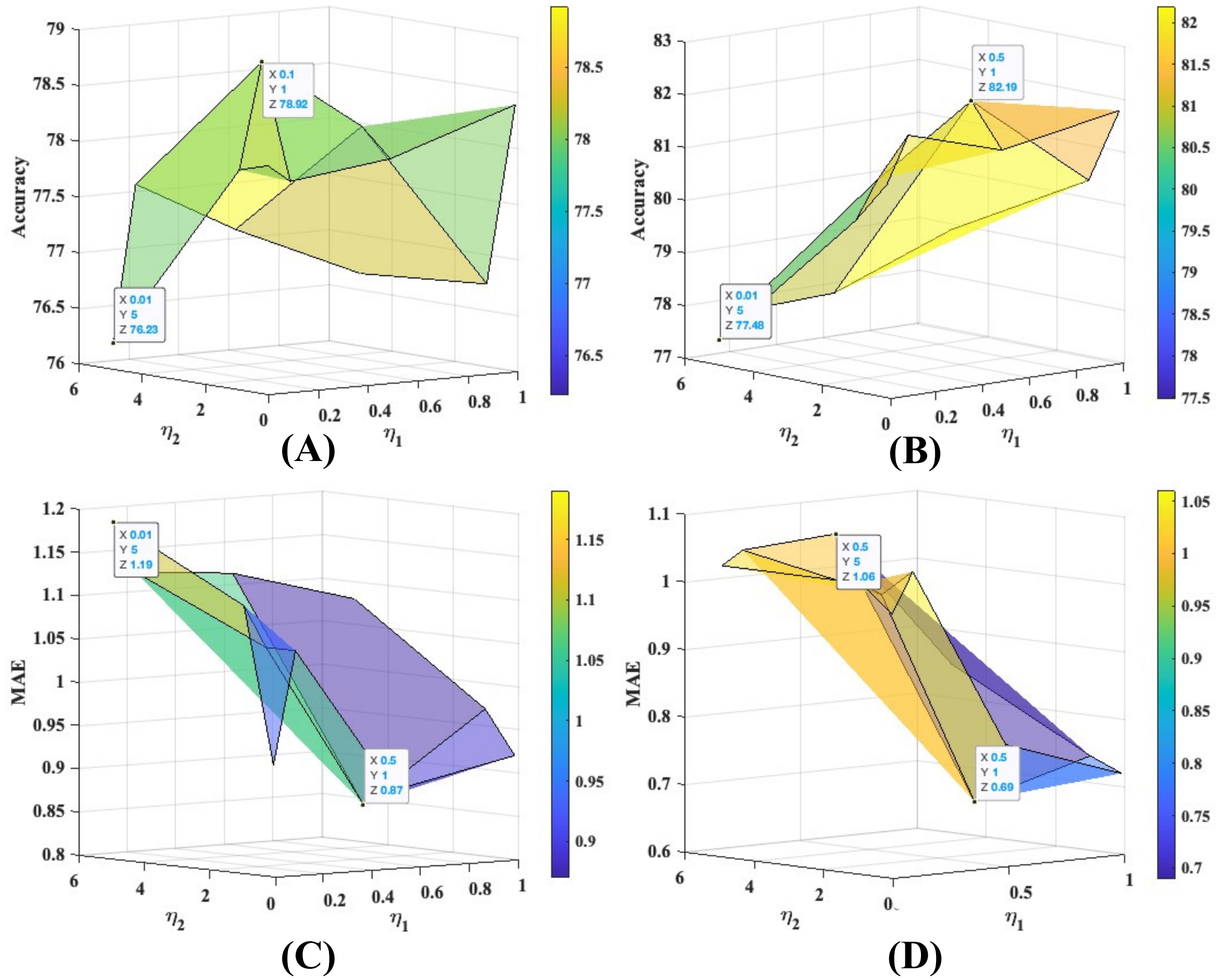


Figure 7: Loss weights analysis. For the AD classification **(A)**, the best $\eta_1 = 0.5, \eta_2 = 1$. For the gender classification **(B)**, the best $\eta_1 = 0.1, \eta_2 = 1$. For the MMSE regression on OASIS **(C)** and HCP **(D)** data, the best $\eta_1 = 0.5, \eta_2 = 1$.

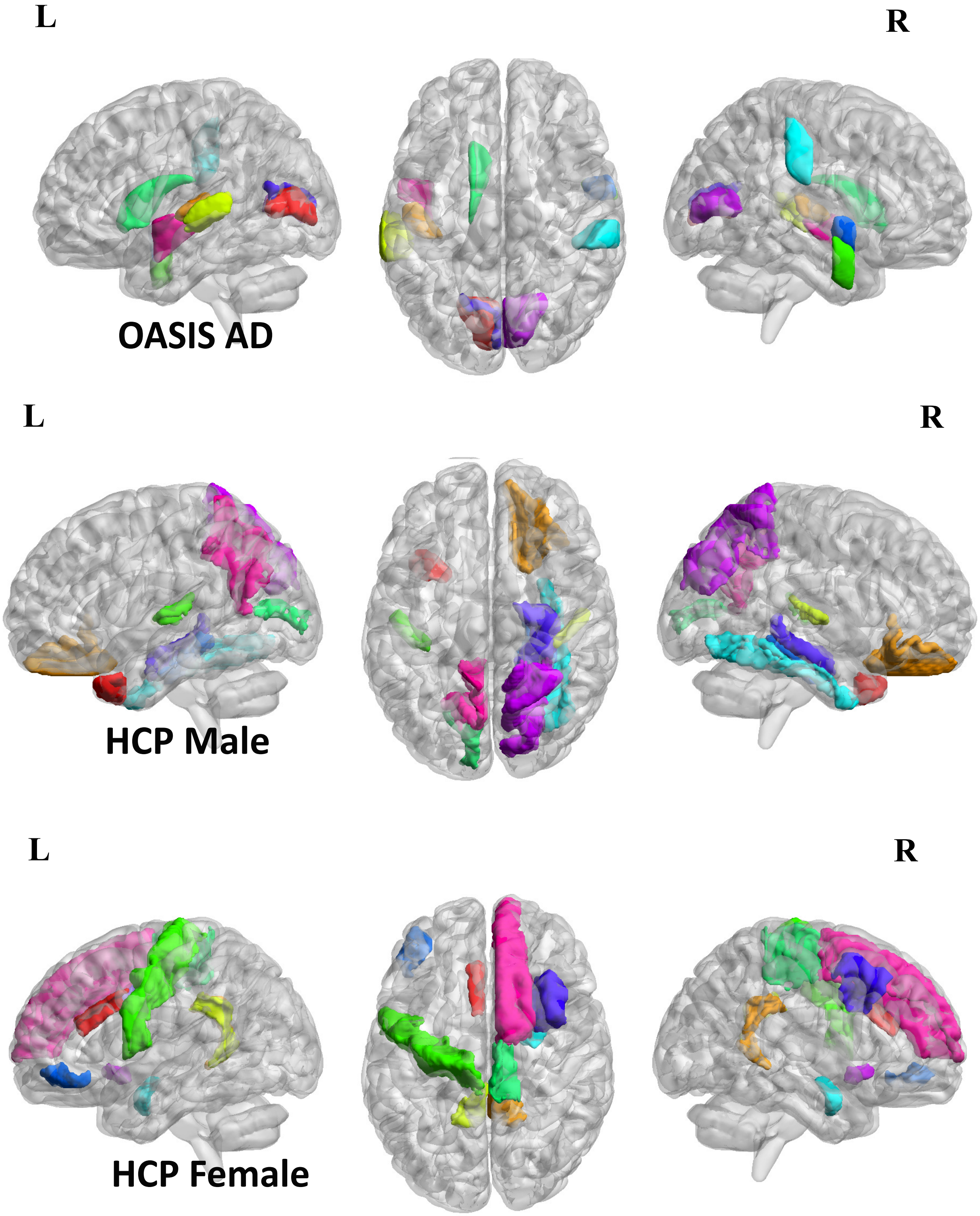


Figure 8: Saliency maps to identify top 10 regions associated with AD (from OASIS) and with each sex (from HCP), respectively

Table 9: The top 10 brain regions in the saliency map

HCP		**OASIS**
Female	Male	AD
R-Superior frontal cortex	L-Precuneus	L-Planum Polare
R-Accumbens-area	R-Superior parietal cortex	R-Intracalcarine Cortex
R-Caudal middle frontal cortex	R-Hippocampus	L-Supracalcarine Cortex
L-Parsorbitalis	R-Parahippocampal	R-Superior Temporal Gyrus, anterior division
R-Amygdala	R-fusiform cortex	R-Supramarginal Gyrus, anterior division
R-Paracentral cortex	L-Pericalcarine	L-Caudate
L-Precentral cortex	L-transverse temporal cortex	R-Middle Temporal Gyrus, anterior division
L-isthmus cingulate	R-transverse temporal cortex	L-Superior Temporal Gyrus, posterior division
R-isthmus cingulate	R-Lateral orbitofrontal	L-Heschl's Gyrus
L-Caudal anterior cingulate	L-Temporal pole	L-Intracalcarine Cortex

4.0 Hierarchical Signed Graph Mining with Contrastive Brain Network Learning

Recently brain networks have been widely adopted to study brain dynamics, brain development and brain diseases. Graph representation learning techniques on brain functional networks can facilitate the discovery of novel biomarkers for clinical phenotypes and neurodegenerative diseases. However, current graph learning techniques have several issues on brain network mining. Firstly, most current graph learning models are designed for unsigned graph, which hinders the analysis of many signed network data (e.g., brain functional networks). Meanwhile, the insufficiency of brain network data limits the model performance on clinical phenotypes predictions. Moreover, few of current graph learning model is interpretable, which may not be capable to provide biological insights for model outcomes. Here, we propose an interpretable hierarchical signed graph representation learning model to extract graph-level representations from brain functional networks, which can be used for different prediction tasks. In order to further improve the model performance, we also propose a new strategy to augment functional brain network data for contrastive learning. We evaluate this framework on different classification and regression tasks using the data from HCP and OASIS. Our results from extensive experiments demonstrate the superiority of the proposed model compared to several state-of-the-art techniques. Additionally, we use graph saliency maps, derived from these prediction tasks, to demonstrate detection and interpretation of phenotypic biomarkers.

4.1 Introduction

Understanding brain organizations and their relationship to phenotypes (e.g., clinical outcomes, behavioral or demographical variables, etc.) are of prime importance in the modern neuroscience field. One of important research directions is to use non-invasive neuroimaging data (e.g., functional magnetic resonance imaging or fMRI) to identify potential imaging biomarkers for clinical purposes. Most previous studies focus on voxel-wise and region-of-interests (ROIs) imaging features [159,161,164]. However, evidences show that the brain is a complex system whose function relies on a diverse set of interactions among brain regions. These brain functions will further determine human clinical or behavioral phenotypes [7,19,20,64,106,116,180,185,218,239]. Therefore, more and more studies have been conducted to predict those phenotypes using the brain network as the delegate of interactions among brain regions [135,173,193]. Additionally, compared to traditional neuroimaging features, brain network has more potential to gain interpretable and system-level insights into phenotype-induced brain dynamics [240]. A brain network is a 3D brain graph model, where graph nodes represent the attributes of brain regions and graph edges represent the connections (or interactions) among these regions.

Many studies have been conducted to analyze brain networks based on the graph theory, however, most of these studies focus on pre-defined network features, such as clustering coefficient, small-worldness [10,21,49,119,206]. This may be sub-optimal since these pre-defined network features may not be able to capture the characteristics of the whole brain network. However, the whole brain network is difficult to be analyzed due to the high dimensionality. To tackle this issue, Graph Neural Network (GNN), as one of embedding techniques, has gained increasing attentions to explore biological characteristics of brain network-phenotype associations in recent years [83, 98, 109]. GNN is a class of deep neural networks that can embed the high-dimensional graph topological structures with graph node features into low dimensional latent space based on the information passing mechanism [102, 199, 221]. A few studies proposed different GNNs to embed the nodes in brain networks and applied a global readout operation (e.g., global mean or sum) to summarize all latent node features as the whole brain network representation for downstream tasks (e.g., behavioral score regression, clinical disease classification) [8, 98, 109, 239]. However, the message passing of GNNs is inherently 'flat' which only propagates information across graph edges and is unable to capture hierarchical structures rooted in graphs which are crucial in brain functional organizations [77,134,137,222]. To address this issue, many recent studies introduce hierarchical GNNs, including node embedding and hierarchical graph pooling strategies, to embed the whole brain network in a hierarchical manner [115, 118, 188, 222, 241].

Although GNNs have achieved great progresses on brain network mining, several issues should be addressed. First, most existing GNNs are designed for unsigned graphs in which all graph nodes are connected via non-negative edges (i.e., edge weights are in the range of $[0, \infty)$). However, signed graphs are very common in brain research (e.g., functional MRI-derived brain networks or brain functional networks), which leads to a demand of signed graph embedding models. To tackle this issue, a few recent studies proposed signed graph embedding models based on the balance-theory [29, 42, 76, 121]. The balance-theory, motivated by human attitudes in social networks, is used to describe the node relationship in signed graphs, where nodes connected by positive edges are considered as 'friends', otherwise are considered as 'opponents'. In the realm of brain functional networks, the positive edge means co-activation and the negative edge indicates anti-activation between those connected nodes. Meanwhile, the balance-theory defines 4 higher-order relationships among graph nodes: (1) the 'friend' of 'friend' is 'friend', (2) the 'opponent' of 'friend' is 'opponent', (3) the 'friend' of 'opponent' is 'opponent', and (4) the 'opponent' of 'opponent' is 'friend'. These definitions are accorded with nodal relationships in the functional brain network, which indicates that the balance theory is applicable in brain functional network embedding. In this study, we adopt the balance theory to co-embed the positive and negative edges as well as local brain nodes. Therefore, generated latent node features include balanced and unbalanced feature components. Beyond focusing on local structures, we also consider the hierarchical structure in graphs as one of global graph features. As suggested by literature [122, 201, 222, 235], graph hierarchical structure can facilitates to yield whole graph representations and to enable the graph-level tasks (i.e., clinical disease classification based on whole brain networks). Particularly, we proposal a new hierarchical pooling module for signed graphs based on the information theory and extend current methods on signed graph from the local embedding to the global embedding.

The second issue is that most of current GNNs on brain network studies are not interpretable, and thus are incapable to provide biological explanations or heuristic insights for model outcomes. This is mainly due to the black-box nature of neural networks. To address this issue, we propose a signed graph learning model with an interpretable graph pooling module. Previous studies indicated that brain networks are hierarchically organized by some regions as neuro-information hubs and peripheral regions, respectively [92, 93, 194, 228]. In our graph pooling module, we compute an information score to measure the information gain for each brain node and choose top-$\boldsymbol{K}$ nodes with high information gains as information hubs. And the information of other peripheral brain nodes will be aggregated onto these hubs. Hence, the proposed pooling module can be interpreted as a brain information hub generator. Apparently, the outcome of this pooling module is a subgraph of the original brain network without creating any new nodes. Therefore, yielded subgraph nodes can be regarded as potential biomarkers to provide heuristic biological explanations for tasks.

To further boost the proposed model performance on prediction tasks, we introduce graph contrastive learning into our proposed hierarchical signed graph representation learning (HSGRL) model. A data augmentation strategy to generate contrastive brain functional network samples is necessary to achieve graph contrastive learning. The data augmentation for contrastive learning aims at creating reasonable data samples, by applying certain transformations, which are similar to original data samples. For example, image rotation and cropping are common transformations to generate new samples in image classification tasks [14, 100, 209, 210, 215]. In graph structural data, a few studies proposed to utilize graph perturbations (i.e., add/drop graph nodes, manipulate graph edges) and graph view augmentation (e.g., graph diffusion) to generate contrastive graph samples from different views [73, 211, 223, 245, 249]. These strategies, although boosting the model performance on large-scale benchmark datasets (e.g., CORA, CITESEER, etc.), may not be suitable to generate contrastive brain network samples. On the one hand, each node in brain networks represents a defined brain region with specific brain activity information so that the brain node can not be arbitrarily removed or added. On the other hand, add/drop operations on brain network may lead to unexpected model outcomes which are difficult to explain and understand from biological views. Motivated by [231, 234], we generate contrastive brain functional network samples directly from fMRI BOLD signals, where the generated contrastive samples are similar to the original ones, and the internal biological structure is therefore maintained. Our main contributions are summarized as follow:

- We propose a hierarchical signed graph representation learning (HSGRL) model to embed brain functional networks and we apply the proposed model on multiple phenotype prediction tasks.

- We propose a contrastive learning architecture with our proposed HSGRL model to boost the model performance on several prediction tasks. A graph augmentation strategy is proposed to generate contrastive samples for fMRI-derived brain network data.
- The proposed HSGPL model is interpretable which yields heuristic biological explanations.
- Extensive experiments are conducted to demonstrate the superiority of our method. Moreover, we draw graph saliency maps for clinical tasks, to enable interpretable identifications of phenotype biomarkers.

4.2 Related Works

4.2.1 Graph Neural Networks and Brain Network Embedding

GNNs are generalized deep learning architectures which are broadly utilized for graph representation learning in many fields (e.g., social network mining [31, 90], molecule studies [40, 47] and brain network analysis [124]). Most existing GNN models (e.g., GCN [102], GAT [199], GraphSage [71]) focus on node-level representation learning and only propagate information across edges of the graph in a flat way. When deploying these models on graph-level tasks (e.g., graph classification, graph similarity learning, [122,131,201,235]), the whole graph representations are obtained by a naive global readout operation (e.g., sum or average all node feature vectors). However, this may lead to poor performance and low efficiency in graph-level tasks since the hierarchical structure, an important property that existed in graphs, is ignored in these models. To explore and capture hierarchical structures in graphs, a few hierarchical graph pooling strategies are proposed to learn representations for the whole graph in a hierarchical manner [62, 115, 222, 224, 241]. Traditional methods to extract brain network patterns are based on graph theory [10, 21, 49, 119, 130, 206] or geometric network optimization [27,105,179,229]. A few recent studies [98,109,238] introduce GNNs to discover brain patterns for phenotypes predictions. However, hierarchical structures in brain networks are not considered in these models, which limits the model performance in a way. Recently, a few hierarchical brain network embedding models are proposed [94, 118, 124].

However, all the aforementioned GNNs are designed for unsigned graph representation learning. A few recent studies are proposed to handle the signed graphs, however, they only consider the node-level representation learning [42, 96, 121, 166]. In this work, we design a signed graph hierarchical pooling strategy to extract graph-level representations from brain functional networks.

4.2.2 Interpretable Graph Learning Model

Generally, the mechanism about how GNNs embed the graph nodes can be explained as a message passing process, which includes message aggregations from neighbor nodes and message (non-linear) transformations [89, 118, 221]. However, most current hierarchical pooling strategies are not interpretable [115, 222, 241]. A few recent studies try to propose interpretable graph pooling strategies to make the pooling module intelligible to the model users. Most of these pooling strategies down-sample graphs relying on network communities which are one of the important hierarchical structures that can be interpreted [38, 118, 132, 188]. For example, [118] proposed a hierarchical graph pooling neural network relying on brain network community to yield interpretable biomarkers. The hierarchical pooling strategy proposed in this work relies on the network information hub which is another important hierarchical structure in brain networks.

4.2.3 Data Augmentation for Graph Contrastive Learning

Most current graph contrastive learning methods augment graph contrastive samples by manipulating graph topological structures. For example, [223, 249] generate the contrastive graph samples by dropping nodes and perturbing edges. Other studies generate contrastive samples by changing the graph local receptive field, which is named as the graph view augmentation [73, 216]. In this work, we introduce the graph contrastive learning into brain functional network analysis and generate contrastive samples from the fMRI BOLD signals.

4.3 Preliminaries of Brain Functional Networks

We denote a brain functional network with N nodes as $G = \{V, E\} = (A, H)$. V is the graph node set where each node (i.e., $v_i, i = 1, ..., N$) represents a brain region. E is the graph edge set where each edge (i.e., $e_{i,j}$) describes the connection between node v_i and v_j. $A \in \mathbb{R}^{N\times N}$ is the graph adjacency matrix where each element, $a_{i,j} \in A$, is the weight of edge $e_{i,j}$. $H \in \mathbb{R}^{N\times C}$ is the node feature matrix where $H_i \in H$ is the $i-th$ row of H representing the feature vector of v_i. Let $B \in \mathbb{R}^{N\times D}$ be the fMRI BOLD signal matrix, where D is the signal length. Generally, the edge weight in the brain functional network can be computed from the fMRI BOLD signal by $a_{i,j} = corr(b_i, b_j)$, where b_i is the $i-th$ row of B representing the BOLD signal of v_i and $corr(\cdot)$ is the correlation coefficient operator. Note that $a_{i,j}$ can be either positive or negative value so that brain functional network is a signed graph. For each subject, we use ˆ and ˇ to denote a functional brain network contrastive sample pair (i.e., $\hat{G} = (\hat{A}, \hat{H})$ and $\check{G} = (\check{A}, \check{H})$).

4.4 Methodology

In this section, we first propose a data augmentation strategy to generate contrastive samples for brain functional networks. Secondly, we introduce our proposed hierarchical signed graph representation learning (HSGRL) model with node embedding and hierarchical graph pooling modules. Finally, we deploy the contrastive learning framework on our proposed HSGRL model to yield the representations for the whole graph, which can be applied to downstream prediction tasks.

4.4.1 Contrastive Samples of Brain Functional Networks

The generation of contrastive samples aims at creating reasonable and similar functional brain network pairs by applying certain transformations. Here we propose a new strategy to generate the brain functional network contrastive samples from fMRI BOLD signals. For each node v_i, we generate two sub-BOLD-signals ($\hat{b}_i$ and $\check{b}_i$) by manipulating its original bold signal b_i. Specifically, we use a window ($size = d$) to clamp the b_i from the signal head and tail, respectively:

$$\begin{aligned} \hat{b}_i &= b_i[d+1, \quad d+2, \quad ..., \quad D] \\ \check{b}_i &= b_i[1, \quad 2, \quad ..., \quad D-d] \end{aligned} \tag{4-17}$$

Obviously, $b_i \in \mathbb{R}^{1\times D}$, $\hat{b}_i$ and $\check{b}_i \in \mathbb{R}^{1\times(D-d)}$. To keep the similarity between $\hat{G}$ and $\check{G}$, we set the window size $d \ll D$. After we generate a pair of sub-bold-signals, we can compute edge weights of the pairwise contrastive brain functional network samples by:

$$\begin{aligned} \hat{a}_{i,j} &= corr(\hat{b}_i, \hat{b}_j) \\ \check{a}_{i,j} &= corr(\check{b}_i, \check{b}_j), \end{aligned} \tag{4-18}$$

where $\hat{a}_{i,j} \in \hat{A}$ and $\check{a}_{i,j} \in \check{A}$ are the weights of $e_{i,j}$ in two contrastive samples. We do not consider the contrastive node features in this work, therefore $\hat{X} = \check{X} = X$. The generated contrastive sample pairs are similar with same node features and slightly different edge weights. We will show this similarity in section 4.5.3.

4.4.2 Hierarchical Signed Graph Representation Learning Model

We present our Hierarchical Signed Graph Representation Learning (HSGRL) model in Figure 9. The HSGRL model includes Balanced and Unbalanced Embedding (BUE) module and Hierarchical Graph Pooling (HGP) module.

4.4.2.1 BUE module

The balance theory is broadly used to analyze the node relationships in signed graphs. The theory states that given a node v_i in a signed graph, any other node (i.e., v_j) can be assigned into either balanced node set or unbalanced node set to v_i regarding to a path between v_i and v_j. Specifically, if the number of negative edges are even in the path between v_i and v_j, then v_j belongs to the balanced set of v_i. Otherwise, v_j belongs to the unbalanced set of v_i. The balance theory indicates that:

- Each graph node, v_j, can belong to either the balanced or unbalanced node set of a given target node v_i.
- The path between v_i and v_j determines the balance attribute of v_j.

Motivated by this, we adopt the idea of signed graph attention networks from [121] to embed brain functional network nodes to generate latent node features with balanced and unbalanced components:

$$X^B, X^U = F_{sign}(A, H) \tag{4-19}$$

where $F_{sign}(\cdot)$ is the signed graph attention encoder [121]. X^B and X^U are the node balanced and unbalanced components of node latent features, respectively. We fuse the two feature components as the node latent features by:

$$X = [X^B \| X^U], \tag{4-20}$$

where [||] denotes concatenate operation.

4.4.2.2 Hierarchical Signed Graph Pooling

As shown in Figure 9, the proposed Hierarchical Graph Pooling (HGP) module consists of 4 steps including: (A) information scores computation, (B) Top-K informative hubs selection, (C) features aggregation and (D) graph pooling.

Information Score Computation: The information score of each node is also considered to contain balanced and unbalanced components to measure the information quantity that each node gains from balanced node set and unbalanced node set, respectively. We first split the signed graph (i.e., with adjacency matrix as A) into positive sub-graph (with adjacency matrix as A_+) and negative one (with adjacency matrix as A_-). Then we utilize Laplace normalization to normalize these two adjacency matrices as:

$$\begin{aligned} \bar{A}_+ &= D_+^{-\frac{1}{2}} A_+ D_+^{-\frac{1}{2}} \\ \bar{A}_- &= D_-^{-\frac{1}{2}} |A_-| D_-^{-\frac{1}{2}}, \end{aligned} \tag{4-21}$$

where $\bar{A}$ is the normalized adjacency matrix. $\boldsymbol{D}_+$ and $\boldsymbol{D}_-$ are degree matrices of A_+ and $|A_-|$, respectively. Note that the i-th line in $\bar{A}$, denoted by $\bar{A}_i$, represents the connectivity probability distribution between v_i and any other nodes. For each node (i.e., v_i), we respectively define the balanced and unbalanced components of information score (IS) by:

$$\begin{aligned} IS_i^B &= \left\|\bar{A}_{+,i:}^\top \otimes X^B\right\|_{\tilde{L}_1} + \left\|\bar{A}_{-,i:}^\top \otimes X^U\right\|_{\tilde{L}_1} \\ IS_i^U &= \left\|\bar{A}_{+,i:}^\top \otimes X^U\right\|_{\tilde{L}_1} + \left\|\bar{A}_{-,i:}^\top \otimes X^B\right\|_{\tilde{L}_1}, \end{aligned} \tag{4-22}$$

where $\|\cdot\|_{\tilde{L}_1}$ is line-wise L_1 norm, and $\otimes$ is the scalar-multiplication between each line of two matrices. $\top$ represents transpose of vector. Then the IS of v_i can be obtained by:

$$IS_i = IS_i^B + IS_i^U. \tag{4-23}$$

Top-K Node Selection and Feature Aggregation: After we obtain the information score for each brain node, we rank the IS and select $\boldsymbol{K}$ brain nodes, with top-$\boldsymbol{K}$ IS values, as informative network hubs. For the other nodes, we aggregate their features on the selected $\boldsymbol{K}$ network hubs based on the feature attention. Particularly, the feature attention between v_i and v_j is computed by: $x_i x_j^\top$. We weighted add (i.e., set feature attentions as weights) the feature of each unselected node to one of hub features, where the attention value between these two nodes is the biggest.

Graph Pooling After the feature aggregation, we down-scale the graph node by removing all unselected nodes. In another word, only the selected top-$\boldsymbol{K}$ network hubs as well as the edges among them will be preserved after graph pooling. Since the functional brain network is a fully connected graph so that no isolated node is existed in the down-scaled graph.

4.4.3 Contrastive Learning Framework with BUE and HGP

The contrastive learning framework with HSGRL is presented in Figure 9. Assume that we forward a pair of contrastive graph samples into the proposed HSGRL model, we will obtain two node latent features, $\hat{X}$ and $\check{X}$ after the last pooling module. We first generate the graph-level representations of two functional brain networks based on the latent node features by a readout operator:

$$\hat{X}_G = \sum_{i=1}^{N'} \hat{x}_i, \quad \check{X}_G = \sum_{i=1}^{N'} \check{x}_i, \tag{4-24}$$

where $\hat{x}_i$ and $\check{x}_i$ are $i-th$ row of $\hat{X}$ and $\check{X}$. $N'(< N)$ is the number of nodes in the down-scaled graph generated by the last pooling module.

4.4.3.1 Contrastive Loss

The normalized temperature-scaled cross entropy loss [170, 195, 213] is utilized to construct the contrastive loss. In the framework training stage, we randomly sample $\boldsymbol{M}$ pairs from the generated contrastive graph samples as a mini-batch and forward them to the proposed HSGRL model to generate contrastive graph representation pairs (i.e., $\hat{X}_G$ and $\check{X}_G$). We use $m \in \{1, ..., M\}$ to denote the ID of the sample pair. The contrastive loss of the $m-th$ sample pair is fomulated as:

$$\ell_m = -log \frac{exp(\Phi(\hat{X}_G^m, \check{X}_G^m)/\alpha)}{\sum_{t=1,t\neq m}^{M} exp(\Phi(\hat{X}_G^m, \check{X}_G^t)/\alpha)}, \tag{4-25}$$

where α is the temperature parameter. $\Phi(\cdot)$ denotes a similarity function that:

$$\Phi(\hat{X}_G^m, \check{X}_G^m) = \hat{X}_G^{m\top} \check{X}_G^m / \|\hat{X}_G^m\| \|\check{X}_G^m\|. \tag{4-26}$$

The batch contrastive loss can be computed by:

$$\mathcal{L}_{contrastive} = \frac{1}{M} \sum_{m=1}^{M} \ell_m \tag{4-27}$$

4.4.3.2 Downstream Task and Loss Functions

We use an MLP to generate the framework prediction for both classification and regression tasks. Specifically, the prediction can be generate by $Y_{pred} = MLP([\hat{X}_G \| \check{X}_G])$. We use $NLLLoss$ and $L_1 Loss$ as supervised loss functions ($\mathcal{L}_{supervised}$) of classification and regression tasks, respectively. The whole framework can be trained in an end-to-end manner by optimizing:

$$\mathcal{L} = \eta_1 \mathcal{L}_{supervised} + \eta_2 \mathcal{L}_{contrastive}, \tag{4-28}$$

where η_1 and η_2 are the loss weights.

4.5 Experiments

4.5.1 Datasets and Data Preprocessing

Two publicly available datasets were used to evaluate our framework. The first includes 1206 young healthy subjects (mean age 28.19 ± 7.15, 657 women) from the Human Connectome Project (HCP) [196]. The second includes 1326 subjects (mean age = 70.42 ± 8.95, 738 women) from the Open Access Series of Imaging Studies (OASIS) dataset [112]. Details of each dataset can be found on their official websites [1] [2]. CONN [207] were used to preprocess fMRI data and the preprocessing pipeline follows our previous publications [3, 59]. For HCP data, each subject's network has a dimension of 82×82 based on 82 ROIs defined using

[1] https://www.oasis-brains.org

[2] https://wiki.humanconnectome.org

FreeSurfer (V6.0) [57]. For OASIS data, each subject's network has a dimension of 132×132 based on the Harvard-Oxford Atlas and AAL Atlas. We deliberately chose different network resolutions for HCP and OASIS to evaluate whether the performance of our new framework is affected by the network dimension or atlas.

4.5.2 Implementation Details

We randomly split the entire functional brain network dataset into 5 disjoint subsets for 5-fold cross-validations in our experiments. The values in the adjacency matrices ($\hat{A}$ and $\check{A}$) of brain functional networks are within range of $[-1, 1]$. We compute the kurtosis and skewness values of the fMRI BOLD signals as the node feature matrices ($\boldsymbol{H}$). We use the Adam optimizer [101] to optimize the loss functions in our model with a batch size of 128. The initial learning rate is $1e^{-4}$ and decayed by $(1 - \frac{current\_epoch}{max\_epoch})^{0.9}$. We also regularized the training with an L_2 weight decay of $1e^{-5}$. We set the maximum number of training epochs as 1000 and, following the strategy in [115], [165], stop training if the validation loss does not decrease for 50 epochs. The experiments were deployed on one NVIDIA RTX A6000 GPU.

4.5.3 Similarities of Contrastive Samples

We utilize the L_1 distance and Cosine Similarity to measure the similarities of the adjacency matrices of contrastive brain networks. Here, we set the window size $d = 10$ to generate the contrastive adjacency matrices. The inner-pair similarity is computed by $\frac{1}{M} \sum_{m=1}^{M} \Psi(\hat{A}^m, \check{A}^m)$, and the inter-pair similarity is computed by $\frac{1}{M^2} \sum_{m=1}^{M} \sum_{t=1}^{M} \Psi(\hat{A}^m, \check{A}^t)$, where $\Psi(\cdot)$ is the similarity function (i.e., L_1 distance or Cosine Similarity). The inner-pair L_1 distances on HCP and OASIS data are 0.1301 and 0.0915, respectively. The inner-pair Cosine Similarities on HCP and OASIS data are 0.9283 and 0.9466, respectively. The inter-pair L_1 distances on HCP and OASIS data are 0.2925 and 0.3137, respectively. The inter-pair Cosine Similarities on HCP and OASIS data are 0.7311 and 0.7014, respectively. We visualize the averaged adjacency matrics on HCP and OASIS data in Figure 10 (A) and (B), respectively, to show their similarities. The original sample is generated by using the whole fMRI BOLD signal (i.e., $d = 0$).

4.5.4 Classification Tasks

4.5.4.1 Experiment Setup

For the comparison, we adopted seven baseline models, which include two traditional graph embedding models (t-BNE [27] and mCCA-ICA [179]), one basic graph neural network (i.e., GCN [102]), two deep graph representation learning models designed for brain network embedding (BrainChey [109] and BrainNet-CNN [98]), and two hierarchical graph neural networks with graph pooling strategies (DIFFPOOL [222] and SAGPOOL [115]). As aforementioned, existing GNN-based models cannot directly take signed graphs as the input, we therefore compute the absolute values of graph adjacency matrices as the input for these baseline models, which is consistent with previous studies [118, 233]. Meanwhile, we compare our model with and without optimizing contrastive loss to demonstrate the effectiveness of contrastive learning in boosting the model performance. The results for gender and Alzheimer Disease (AD) classification are reported in accuracy, precision and F1-score with their standard deviation (*std*). The results for zygosity classification (i.e., 3 classes classification task with class labels as: not twins, monozygotic twins and dizygotic twins) are reported in accuracy and Macro-F1-score with their *std*. The number of cascaded BUE and HGP modules are set to 3 and the number of top-K nodes in the pooling module is 50% of the number of nodes in the current graph. We search the loss weights η_1 and η_2 in range of $[0.1, 1, 5]$ and $[0.01, 0.1, 0.5, 1]$ respectively and determine the loss weights as $\eta_1 = 1$, $\eta_2 = 0.1$. The temperature parameter in contrastive loss is set as 0.2. Details of the hyperparameters analysis are shown in section 4.5.6.

4.5.4.2 Results

Table 10 shows the results of gender classification, zygosity classification and AD classification. It shows that our model achieves the best performance comparing to all baseline methods on three tasks. For example, in the gender classification, our model outperforms the baselines with at least 8.56%, 8.18% and 8.91% increases in accuracy, precision and F1 scores, respectively. In general, the deep graph neural networks are superior than the traditional graph embedding methods (i.e., t-BNE and mCCA-ICA). When we remove the supervision of the contrastive loss, the performance, though comparable to baselines, decreases in a way. This manifests the effectiveness of the contrastive learning which can substantially boost the model performance.

4.5.5 Regression tasks

4.5.5.1 Experiment Setup

In the regression tasks, we use the same baselines for comparisons. The regression tasks include predicting MMSE scores on OASIS data, Flanker scores, Card-Sort scores, and 3 ASR scores (i.e., Aggressive, Intrusive and Rule-Break scores) on HCP data. Particularly, MMSE (Mini-Mental State Exam) test [189], Flanker test [50] and Wisconsin Card-Sort test [13, 141, 148] are 3 neuropsychological tests designed to measure the status and risks of human neurodegenerative disease and mental illness. The ASR (Achenbach Adult Self-Report) is a life function which is used to measure the emotion and social support of adults. The structure of proposed model remains unchanged. The loss weights are set as $\eta_1 = 0.5$ and $\eta_2 = 1$. The regression results are reported in average Mean Absolute Errors (MAE) with its *std* under 5-fold cross validations.

4.5.5.2 Results

The regression results are presented in Table 11. It shows that our model achieves the best MAE values comparing to all baseline methods. Similar to the classification tasks, the deep graph neural networks are superior than traditional graph embedding methods (i.e., t-BNE and mCCA-ICA). Comparing our method with and without the supervision of the contrastive loss, we can hold the conclusion that the contrastive learning can further boost the model performance.

4.5.6 Ablation Studies

In this section, we investigate the effect of 4 hyperparameters on our model performance, including (1) the window size (d) which we used to clamp the fMRI BOLD signals when generating contrastive functional brain network samples, (2) temperature parameter (α) within contrastive loss, (3) the number of BUE and HGP modules utilized in HSGRL model, and (4) loss weights η_1 and η_2. First, we set the window size as $[0, 5, 10, 20, 30, 40, 50]$, respectively and generate different contrastive samples as the input of our proposed model. The first column in Figure 11 shows the analysis of the window size parameter. It indicates that the best window size is around $d = 10$. When the window size decreases to 0, the model performance declines since the data is only duplicated without any substantial new samples. It is interesting that the performance when $d = 0$ is even worse than that obtained without contrastive learning but with contrastive samples generated with $d = 10$ (see Ours w/o Contrastive in Table 10 and 11). The reason is that data augmentation is introduced in the latter case but not in the first case. Second, we increase the temperature α from 0.1 to 1.0 with a step of 0.1. The second column in Figure 11 demonstrates the analysis of the temperature parameter. It shows that the best temperature value for our framework is $\alpha = 0.2$. Moreover, we set the number of BUE and HGP modules as $[1, 2, 3, 4, 5]$, respectively for our framework. The third column in Figure 11 shows the analysis of this parameter. It manifests that the framework performance is consistent and steady when different number of BUE and HGP modules are deployed. The best number of the modules for almost all tasks are 3, except for the regression tasks on Flanker and Aggressive. Finally, we present the loss weights analysis (see Figure 12) on the 3 classification tasks and the best results are achieved when $\eta_1 = 1$ and $\eta_2 = 0.1$.

4.5.7 Interpretation with Brain Saliency Map

Within our new graph pooling module, an information score is designed to measure the information gain for each brain node and only top-K nodes with high information gains will be preserved as brain information hubs while the information of other peripheral nodes will be aggregated onto these hubs. These hubs, through the final pooling layer, will serve as the delegate of the whole brain network and then be linked to clinical phenotypes (e.g., clinical/behavior scores or diagnosis). Therefore, they can provide hints for further clinical analyses on how this phenotype is associated with brain functional network from the global view. We utilize the Class Activation Mapping (CAM) approach [6, 151, 237] to generate the brain network saliency map, which indicates the top brain regions associated with each prediction task. Figures 13 and 14 illustrate Brain Saliency Maps for classification and regression tasks, respectively. For example, in the classification task (AD vs. NC), the saliency map for AD highlights multiple regions (such as Planum Polare, Frontal Operculum cortex, Supracalcarine Cortex, etc.) which are conventionally conceived as the biomarkers of AD in medical imaging analysis [70, 78, 110, 156]. In the meantime, the saliency map for NC highlights many regions in Cerebellum and Frontal lobe. These regions control cognitive thinking, motor control, and social mentalizing as well as emotional self-experiences [162,175, 197], in which AD patients typically show problems. Another example is the classification of Male vs. Female on HCP data. Females are more ”emotional” or ”sensitive”, suggested by the regions such as isthmuscingulate and caudalanteriorcingulate while males tend to be more competitive and dominant, manifested in regions such as lateralorbitofrontal and precuneus. These results are consistent with previous findings in the literature [1, 25, 35, 80]. The details of all highlighted brain regions in each task are summarized in the Table 12 for OASIS dataset, and in the Table 13a and Table 13b for HCP dataset. These highlighted regions can help us locating brain regions associated with any phenotype, which provide clues for future clinical investigations.

4.6 Conclusion

We propose a novel contrastive learning framework with an interpretable hierarchical signed graph representation learning model for brain functional network mining. Additionally, a new data augmentation strategy is designed to generate the contrastive samples for brain functional network data. Our new framework is capable of generating more accurate representations for brain functional networks compared to other state-of-the-art methods and these network representations can be used in various prediction tasks (e.g., classification and regression). Moreover, Brain saliency maps may assist with phenotypic biomarker identification and provide interpretable explanation on framework outcomes.

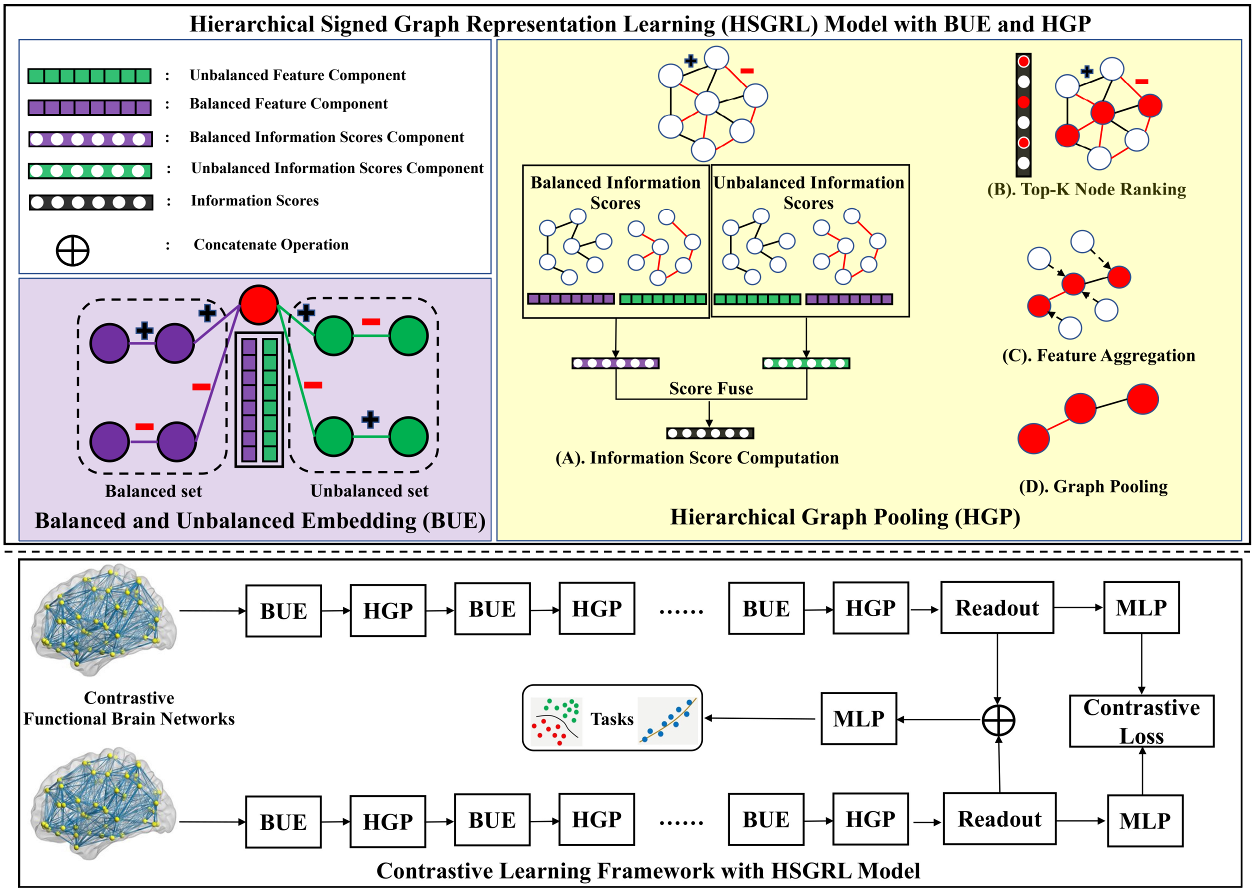

Hierarchical Signed Graph Representation Learning (HSGRL) Model with BUE and HGP
: Unbalanced Feature Component
: Balanced Feature Component
: Balanced Information Scores Component
: Unbalanced Information Scores Component
: Information Scores
: Concatenate Operation
Balanced set
Unbalanced set
Balanced and Unbalanced Embedding (BUE)
Balanced Information Scores
Unbalanced Information Scores
Score Fuse
(A). Information Score Computation
(B). Top-K Node Ranking
(C). Feature Aggregation
(D). Graph Pooling
Hierarchical Graph Pooling (HGP)
Contrastive Functional Brain Networks
BUE
HGP
......
Readout
MLP
Contrastive Loss
Tasks
Contrastive Learning Framework with HSGRL Model

Figure 9: Diagram of the proposed contrastive graph learning framework (in the bottom black box) with hierarchical signed graph representation learning (HSGRL) model (in the top black box) for functional brain network embedding and downstream tasks (i.e., phenotype classification or regression). The HSGRL model consists of cascaded BUE and HGP modules to extract graph-level representations of contrastive brain functional network pairs (i.e., $\hat{X}_G$ and $\check{X}_G$) in a hierarchical manner. The $\hat{X}_G$ and $\check{X}_G$ participate to build up the contrastive loss for graph constrastive learning. Meanwhile, a concatenate operation is utilized to generate the fused graph feature by $X_G = [\hat{X}_G \| \check{X}_G]$). The fused graph feature X_G is utilized for downstream prediction tasks (i.e., graph classification and regression).

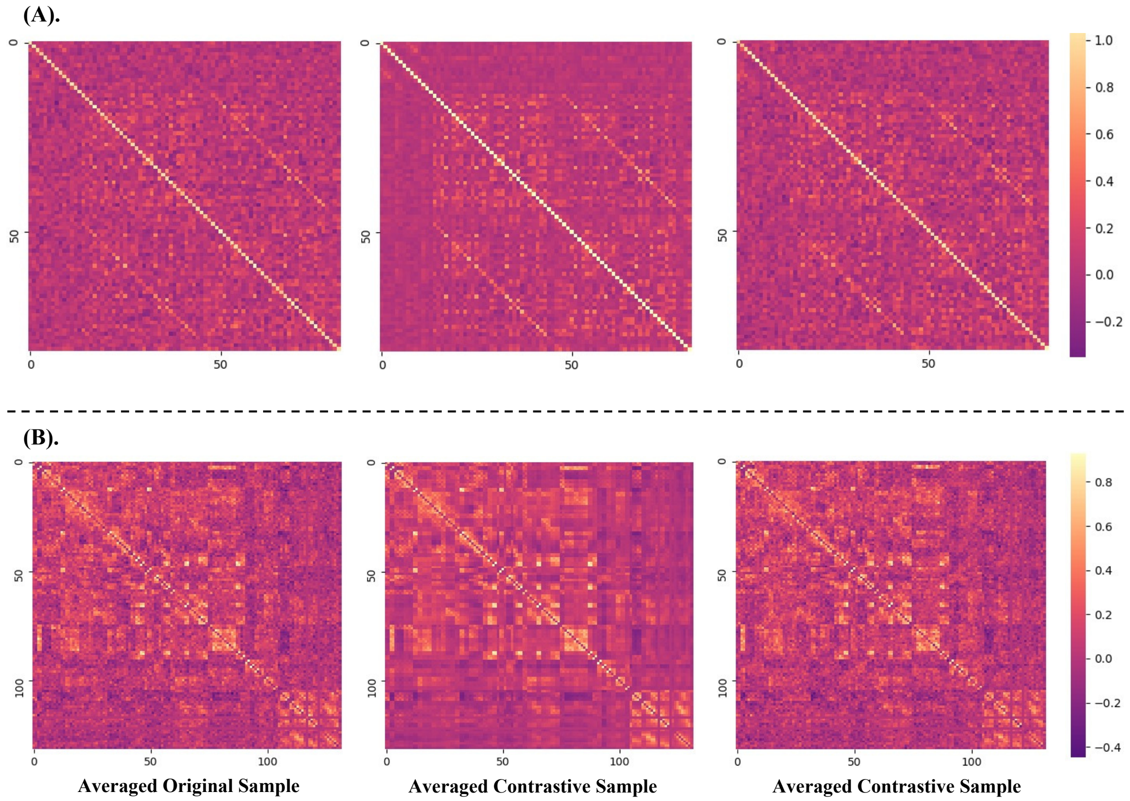


Figure 10: Visualization of the averaged adjacency matrices for original and contrastive samples on **(A).** HCP dataset and **(B).** OASIS dataset. The averaged contrastive sample pair is generated by using a window size $d = 10$.

Table 10: Classification accuracy with s.t.d values under 5-fold cross-validation on gender classification, zygosity classification and AD classification tasks. The values in **bold** show the best results.

Method	**HCP**					**OASIS**		
	Gender			Zygosity		AD		
	Acc.	Pre.	F1.	Acc.	Macro-F1.	Acc.	Pre.	F1.
t-BNE	63.84(2.09)	64.17(1.90)	63.264(2.12)	37.19(2.65)	39.67(3.04)	61.26(2.31)	63.58(2.06)	62.05(1.97)
mCCA-ICA	61.21(4.03)	63.11(3.75)	62.20(3.59)	35.51(4.64)	38.71(3.34)	63.37(1.98)	62.06(2.12)	64.37(2.09)
GCN	66.76(2.22)	65.09(3.13)	67.58(2.84)	46.66(2.14)	47.21(2.51)	67.37(2.69)	69.21(2.00)	68.51(4.29)
SAGPOOL	68.12(3.07)	69.96(2.48)	67.51(2.65)	49.91(2.22)	51.07(2.31)	67.23(2.15)	68.83(1.13)	67.51(2.51)
DIFFPOOL	72.06(2.28)	74.05(1.90)	73.07(2.42)	53.37(1.88)	54.28(2.14)	72.79(1.66)	71.55(2.15)	70.83(2.01)
BrainCheby	75.08(1.98)	76.14(2.38)	74.09(1.84)	56.25(2.12)	57.37(2.05)	72.55(2.45)	73.36(1.88)	72.62(1.33)
BrainNet-CNN	74.09(2.49)	73.71(1.96)	73.27(2.21)	54.03(2.20)	55.25(2.46)	68.37(1.71)	69.97(1.30)	68.51(2.02)
Ours w/o Contrastive	**78.86(2.18)**	**80.06(1.33)**	**77.52(1.69)**	**61.05(1.70)**	**63.24(2.51)**	**76.26(2.32)**	**75.42(1.62)**	**76.80(1.72)**
Ours	**81.51(1.14)**	**82.37(1.95)**	**80.69(2.03)**	**63.33(2.06)**	**64.51(1.74)**	**77.51(1.84)**	**78.83(1.78)**	**78.28(1.95)**

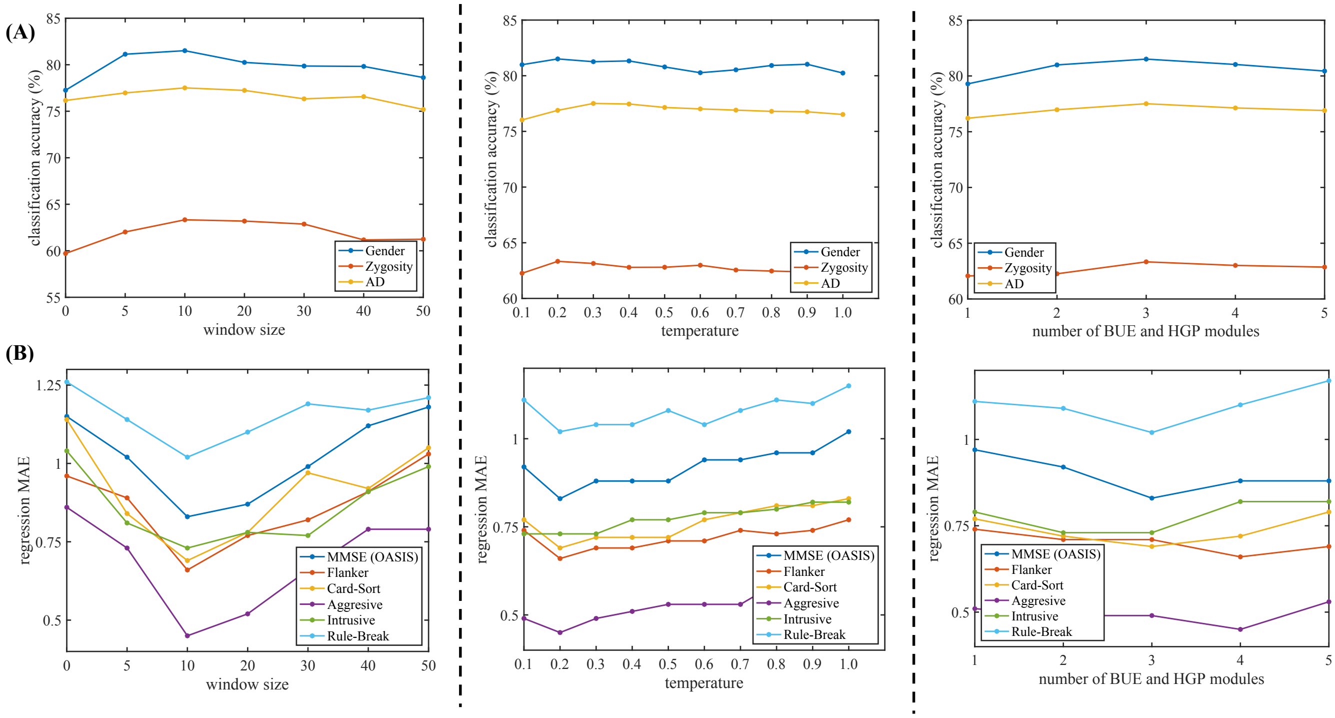


Figure 11: Parameter analysis. The model performance obtained with: contrastive samples generated by different window sizes (Column 1), different temperature parameters in contrastive loss (Column 2), and different number of BUE and HGP modules (Column 3). **(A)** shows the analysis on classification tasks and **(B)** shows the analysis on regression tasks.

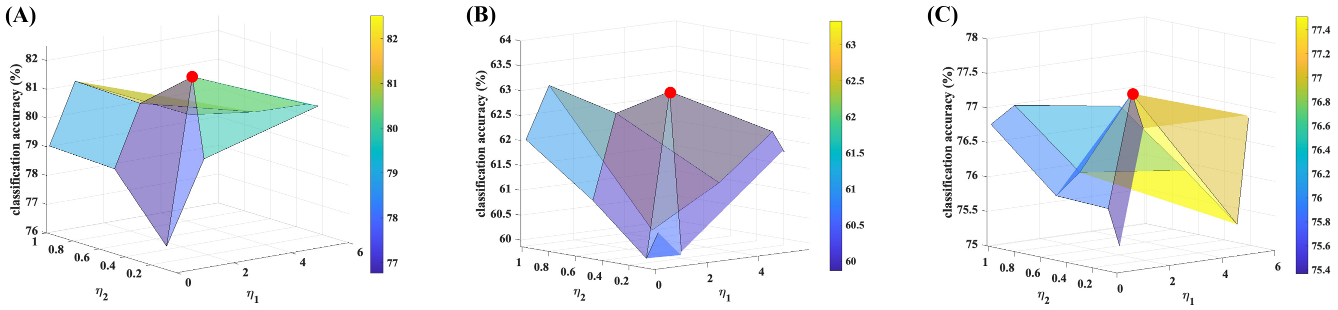


Figure 12: Loss weights analysis on classification tasks. **(A)** shows the analysis on gender classification, **(B)** shows the analysis on zygosity classification and **(C)** shows the analysis on AD classification. The red points represent the best results, where $\eta_1 = 1$ and $\eta_2 = 0.1$.

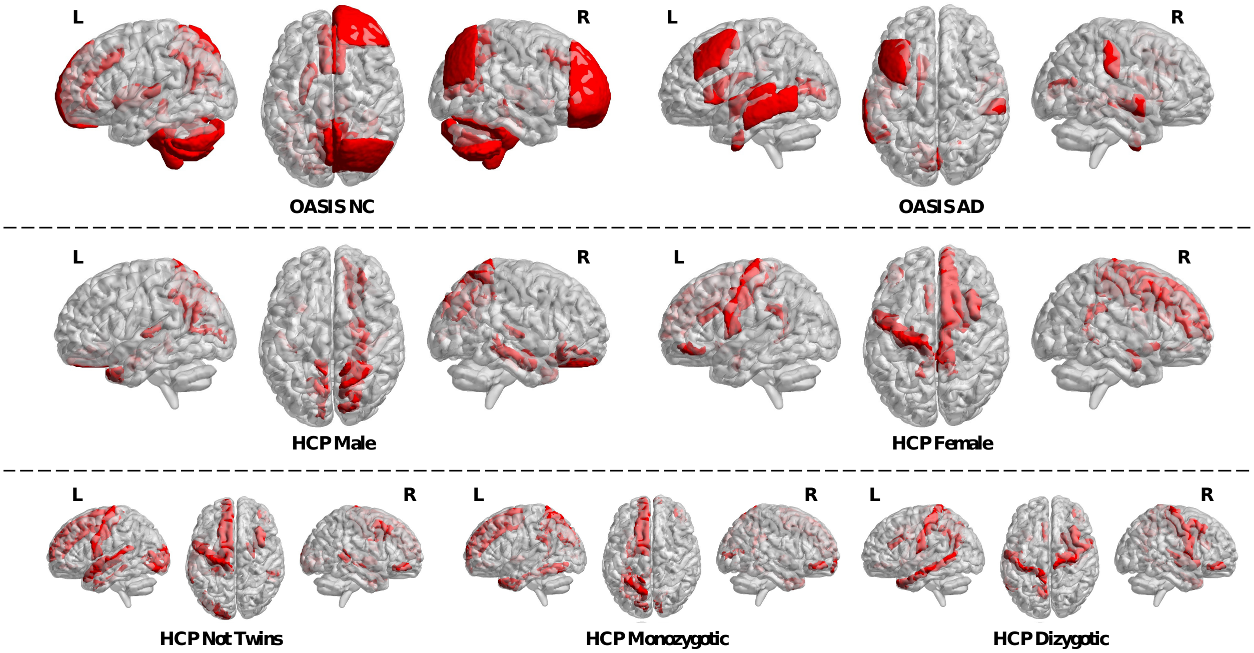


Figure 13: Brain saliency maps for classification tasks. Here we identify: (1) top 15 regions associated with AD and NC from OASIS, (2) top 10 regions associated with each sex and each zygosity from HCP.

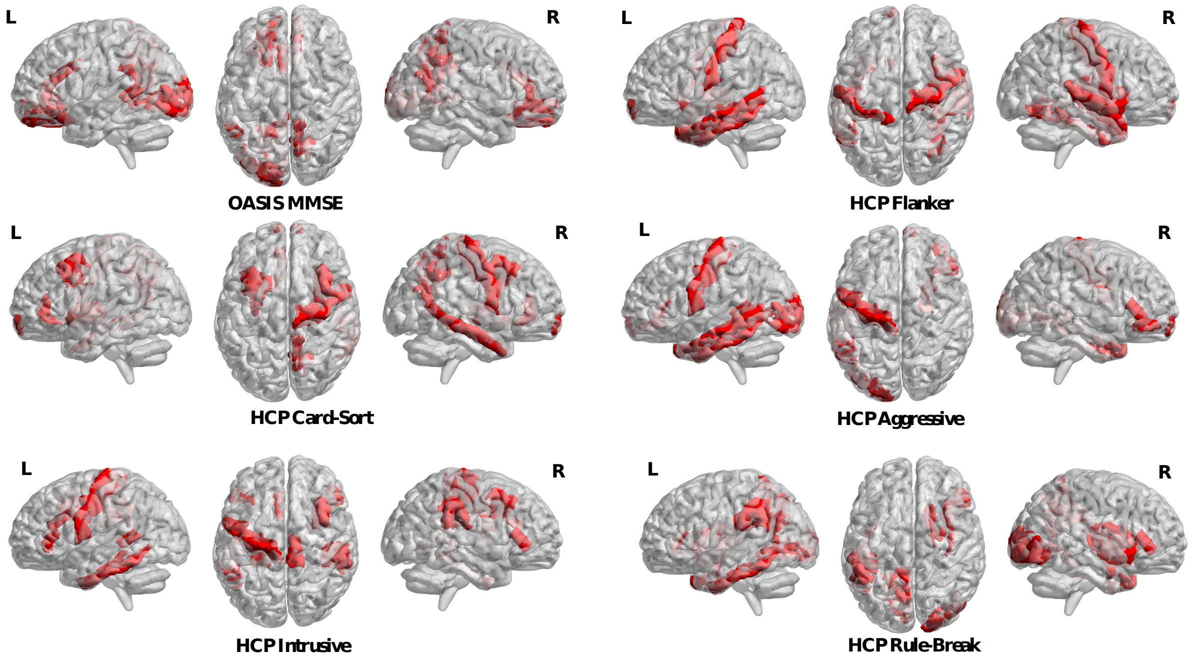


Figure 14: Brain saliency maps for regression tasks. Here we identify: (1) top 15 regions associated with MMSE from OASIS, (2) top 10 regions associated with Flanker score, Card-Sort score, Aggressive score, Intrusive score and Rule-Break score from HCP.

Table 11: Regression Mean Absolute Error (MAE) with s.t.d under 5-fold cross-validation. The values in **bold** show the best results.

Method	**OASIS**	**HCP**				
	MMSE	Flanker	Card-Sort	Aggressive	Intrusive	Rule-Break
t-BNE	2.02(0.36)	1.69(0.19)	1.58(0.22)	1.89(0.10)	1.84(0.22)	1.77(0.41)
mCCA-ICA	2.68(0.19)	1.82(0.21)	1.67(0.17)	1.47(0.26)	1.97(0.13)	1.61(0.29)
GCN	2.05(0.07)	1.67(0.15)	1.46(0.11)	1.59(0.32)	1.66(0.24)	1.69(0.08)
SAGPOOL	1.84(0.33)	1.55(0.06)	1.44(0.13)	1.52(0.18)	1.50(0.24)	1.74(0.23)
DIFFPOOL	1.27(0.20)	1.34(0.14)	1.16(0.30)	1.27(0.41)	1.25(0.07)	1.43(0.15)
Brain-Cheby	1.51(0.67)	1.17(0.26)	1.24(0.31)	0.79(0.06)	1.09(0.21)	1.58(0.41)
BrainNetCNN	1.26(0.19)	1.43(0.24)	**0.91(0.11)**	1.33(0.23)	1.14(0.13)	1.29(0.19)
Ours w/o Contrastive	**1.02(0.11)**	**0.89(0.13)**	0.97(0.20)	**0.74(0.17)**	**0.96(0.15)**	**1.15(0.11)**
Ours	**0.83(0.24)**	**0.66(0.17)**	**0.69(0.14)**	**0.45(0.12)**	**0.73(0.08)**	**1.02(0.16)**

Table 12: The list of highlighted brain regions for OASIS dataset, including AD and NC classification tasks and MMSE regression task.

AD	Planum Polare Left	Frontal Operculum Cortex Left	Supracalcarine Cortex Left	Left-Caudate	Supramarginal Gyrus, anterior division Right	Superior Temporal Gyrus, anterior division Right	Middle Temporal Gyrus, posterior division Left	Superior Temporal Gyrus, posterior division Left
	Heschl's Gyrus Left	Intracalcarine Cortex Left	Middle Frontal Gyrus Left	Planum Polare Right	Temporal Fusiform Cortex, anterior division Left	Middle Temporal Gyrus, temporooccipital part Left	Supracalcarine Cortex Right	—
NC	Paracingulate Gyrus Right	Intracalcarine Cortex Right	Frontal Pole Right	Cerebelum 6 Right	Paracingulate Gyrus Left	Left-Putamen	Cerebelum 8 Left	Cerebelum 7b Right
	Heschl's Gyrus Left	Cuneal Cortex Right	Precuneous Cortex	Cerebelum Crus2 Left	Lateral Occipital Cortex, superior division Right	Brain-Stem	Cerebelum 8 Right	—
MMSE	Right-Caudate	Temporal Pole Right	Planum Temporale Left	Cerebelum Crus1 Right	Middle Temporal Gyrus, posterior division Right	Temporal Occipital Fusiform Cortex Left	Temporal Occipital Fusiform Cortex Right	Middle Temporal Gyrus, temporooccipital part Left
	Planum Temporale Right	Frontal Orbital Cortex Left	Vermis 9	Temporal Pole Left	Middle Temporal Gyrus, temporooccipital part Right	Left-Caudate	Temporal Pole Left	—

Table 13a: The list of highlighted brain regions for classification tasks on HCP Dataset.

Male	Female	Not Twins	Monozygotic	Dizygotic
ctx-lh-precuneus	ctx-rh-superiorfrontal	ctx-lh- lateraloccipital	ctx-lh- isthmuscingulate	ctx-lh-postcentral
ctx-rh- superiorparietal	Right-Accumbens-area	ctx-rh-bankssts	ctx-rh-pericalcarine	ctx-rh- transversetemporal
Right-Hippocampus	ctx-rh- caudalmiddlefrontal	ctx-lh-precentral	ctx-rh-frontalpole	ctx-rh- transversetemporal
ctx-rh- parahippocampal	ctx-lh-parsorbitalis	ctx-lh- parahippocampal	ctx-lh-fusiform	Paracingulate Gyrus Right Paracingulate
Right-Amygdala	Right-Amygdala	ctx-lh-entorhinal	ctx-lh-entorhinal	ctx-lh-caudalanteriorcingulate
ctx-lh-pericalcarine	ctx-rh-paracentral	Right-Pallidum	ctx-lh- superiorfrontal	ctx-rh-parsorbitalis Right-Putamen
ctx-lh-transversetemporal	ctx-lh-precentral	ctx-lh- superiortemporal	ctx-lh- temporalpole	ctx-rh-precentral
ctx-rh-transversetemporal	ctx-lh- isthmuscingulate	ctx-rh-parsorbitalis	ctx-lh- superiorparietal	ctx-rh-caudalmiddlefrontal
ctx-rh- lateralorbitofrontal	ctx-rh- isthmuscingulate	ctx-lh-superiorfrontal	Left-Pallidum	ctx-lh-precuneus
ctx-lh- temporalpole	ctx-lh- caudalanteriorcingulate	ctx-rh- caudalmiddlefrontal	ctx-rh-parsorbitalis	ctx-lh-temporalpole

Table 13b: The list of highlighted brain regions for regression tasks on HCP Dataset.

Flanker	**Card-Sort**	**Aggressive**	**Intrusive**	**Rule-Break**
Left-Accumbens-area	Left-Accumbens-area	ctx-lh-bankssts	ctx-lh-bankssts	ctx-lh-precuneus
ctx-lh-inferiortemporal	ctx-lh- caudalmiddlefrontal	ctx-lh- inferiortemporal	ctx-lh- inferiortemporal	ctx-lh- inferiortemporal
ctx-rh-insula	ctx-rh-frontalpole	ctx-lh- lateraloccipital	ctx-lh- parahippocampal	Right-Caudate
ctx-lh- middletemporal	ctx-lh- rostralanteriorcingulate	ctx-lh-precentral	ctx-rh- supramarginal	ctx-rh- lateraloccipital
ctx-lh-postcentral	ctx-rh- middletemporal	ctx-rh-frontalpole	ctx-rh-paracentral	ctx-lh- supramarginal
ctx-lh-temporalpole	ctx-lh-frontalpole	ctx-rh-parsorbitalis	ctx-rh- parstriangularis	ctx-rh-insula
ctx-rh- superiortemporal	ctx-rh-precentral	ctx-rh- parstriangularis	ctx-lh- caudalanteriorcingulate	ctx-rh- parstriangularis
ctx-lh-frontalpole	ctx-rh- caudalmiddlefrontal	ctx-lh- middletemporal	ctx-lh-precentral	ctx-lh-lingual
ctx-rh-precentral	ctx-rh-precuneus	ctx-rh-entorhinal	ctx-rh- caudalmiddlefrontal	ctx-rh- temporalpole
ctx-rh-fusiform	Left-Putamen	ctx-rh-temporalpole	ctx-lh-parsorbitalis	Right-Amygdala

5.0 Graph Reasoning for Semantic Segmentation

The COVID-19 pandemic has extremely threatened human health, and automated algorithms are needed to segment infected regions in the lung using computed tomography (CT). Although several deep convolutional neural networks (DCNNs) have proposed for this purpose, their performance on this task is suppressed due to the limited local receptive field and deficient global reasoning ability. To address these issues, we propose a segmentation network with a novel pixel-wise sparse graph reasoning (PSGR) module for the segmentation of COVID-19 infected regions in CT images. The PSGR module, which is inserted between the encoder and decoder of the network, can improve the modeling of global contextual information. In the PSGR module, a graph is first constructed by projecting each pixel on a node based on the features produced by the encoder. Then, we convert the graph into a sparsely-connected one by keeping $\boldsymbol{K}$ strongest connections to each uncertainly segmented pixel. Finally, the global reasoning is performed on the sparsely-connected graph. Our segmentation network was evaluated on three publicly available datasets and compared with a variety of widely-used segmentation models. Our results demonstrate that (1) the proposed PSGR module can capture the long-range dependencies effectively and (2) the segmentation model equipped with this PSGR module can accurately segment COVID-19 infected regions in CT images and outperform all other competing models.

5.1 Introduction

The pandemic of COVID-19 has become one of the most severe global health crises in human history, leading to enormous loss of population and prosperity [202]. Although it has been well recognized as the gold standard for COVID-19 screening, the reverse transcription polymerase chain reaction (RT-PCR) requires a huge amount of human resources and medical equipment and is limited by its high false negative rate [2, 52]. Thanks to the development of computer-aided diagnose (CAD) techniques, radiological imaging modalities have been integrated into an uniform platform to detect and diagnose the disease [86]. Among them, computed tomography (CT) has been broadly utilized as an assistance to RT-PCR due to its superior imaging quality and 3-dimensional view of the lungs [2,52].

Beyond screening COVID-19 cases, the segmentation of COVID-19 infection using CT can benefit the prediction of the pathological stage, development, and treatment response of the disease. Previously, the segmentation is usually conducted by radiologists through visual inspections, which is time consuming, professional-skill intensive. Deep convolutional neural networks (DCNNs) are mainly constructed with convolutional filters which can learn and extract abundant high-dimensional features for efficient vision understanding in an end-to-end fashion. Since the AlexNet [108] showed excellent results in the ImgaeNet 2012 Challenge [160], DCNNs have been widely used in various of computer vision tasks and have improved the performance substantially [39,203,205].

The success of DCNNs in computer vision society has also prompted investigators to apply DCNNs to COVID-19 infection segmentation in CT images [5,51,155,212,217,246]. Despite several attempts, this segmentation task, however, remains challenging due to the fact that COVID-19 infected regions usually (1) vary in shape, size, and location; (2) appear to be visually similar to their surroundings tissues; and (3) disperse wildly within the lung cavity. We believe that the segmentation performance of current DCNN-based solutions tends to be suppressed by their limited local receptive fields, which result in an insufficient ability in modeling the long-range dependency.

In recent years, the non-local methods, which introduce the self-attention mechanism [198] to DCNNs, have been proposed to enhance the long-range dependency of spatial contextual information for semantic segmentation [61, 91, 204, 243]. Although being able to capture, to some extent, the long-range information, these methods usually construct excessive correlations among all pixels, which may introduce redundant information and suppress the discriminatory power of image features. Moreover, the global reasoning ability of these models is still limited, since the interactive information is merely delivered and aggregated at the image level.

Recently, graph neural networks (GNNs) have enjoyed increasing success and advanced ever more powerful in semantic image segmentation, showing great potentials in enhancing DCNNs with the global reasoning ability. GNNs are designed to concert an image into a graph, with pixels and pixel interactions being represented by nodes and edges, respectively. In general, GNNs include three operations: information propagation, information aggregation, and feature transformation [221]. Based on the information propagation mechanism, any node in a graph can gain the information across all nodes from the whole graph [65, 71, 163, 200]. Thus, GNNs can break through the restriction of the local receptive field of the traditional convolution filter and enables the long-range dependency reasoning based on the global feature map.

When incorporating a GNN into a DCNN-based image segmentation model, a crucial step is to construct a projection that maps DCNN-generated features to the graph space. Usually, a cluster of pixels is identified based on feature similarity and is directly projected onto a graph node [36, 120]. This projection scheme requires a predetermined number of nodes, which may not be suitable for all cases. Alternatively, an image can be partitioned into regions based on its structural information or pseudo landmarks, and each region is then projected onto a node [125]. This solution relies highly on prior knowledge for image partitioning and has poor generalizability. Particularly, the COVID-19 infected regions in CT images are usually small, disperse, and morphologically diverse, whereas normal regions are usually large [51]. When using the cluster- or region-based method to convert a COVID-19 CT image into a graph, a huge number of normal pixels might be projected onto one node, which contains too much diverse information that could suppress other nodes, which represent small infected regions, in the graph reasoning stage. Moreover, due to the inaccuracy of pixel clustering or image partitioning, both projection schemes may assign a few normal pixels to a node that represents an infected region and vice versa. Such inaccuracy may disturb subsequent pixel classification and can hardly be corrected in the subsequent stage. To address these drawbacks, an intuitive solution is to map each pixel to a node, which, similar to those non-local methods, would build a densely connected graph. However, such a solution may be intractable due to its extremely high computational and spatial complexity. Moreover, it is not reasonable and necessary for each node to gain effective information from all other nodes, since extra noise may be introduced during this process [81].

In this paper, we propose a convolutional neural network with pixel-wise sparse graph reasoning (PSGR) for the segmentation of COVID-19 infection in chest CT images. The PSGR module is inserted between the encoder and the decoder of the network. The workflow of PSGR consists of three steps. First, a densely-connected graph is constructed by projecting each pixel onto a node based on the features generated by the segmentation backbone. Second, the graph is converted into a sparsely-connected one by keeping only $\boldsymbol{K}$ strongest connections to each uncertain node (pixel). The strength of a connection is measured by the similarity of two nodes in the feature space and the uncertainty of each node is determined by an additional coarse segmentation branch. Third, the long-range information reasoning is performed on the sparsely-connected graph and the enhanced features are generated and fed to the segmentation backbone. The theoretical contribution of the work is that we propose a method, based on the information theory, to yield a sparse graph from the dense counterpart, which facilitates the effective information propagation and suppresses noise across graph nodes. Ideally, we consider that a graph node which gains much information from other nodes is informative, and the graph connection (i.e., a graph edge) between two informative nodes can propagate the information effectively. Whether a node in the constructed graph is informative with a high information gain is determined by our proposed node information score. Details will be discussed in Section 5.4.1.1. The proposed solution has been evaluated against widely-used segmentation models on three public CT datasets for both bi-class and multi-class COVID-19 infection segmentation.

The main contributions are summarized as follows:

- We propose a new approach to enhance the modeling of long-range dependencies for COVID-19 infected region segmentation in CT images, in which the image features are projected pixel-wisely to the graph space for global reasoning.
- An edge pruning method is developed to convert the graph into a sparsely-connected one, resulting in effective information retrieval and reduced noise propagation among nodes.
- The results on three datasets indicate that the segmentation network with the proposed PSGR module is superior to all competing models in the segmentation of COVID-19 infected regions in CT images, and the ablation study also demonstrates the effectiveness of our PSGR module.

The remaining sections of this paper are organized as follows. Some related works are summarized in Section 5.2. The preliminaries of graph neural network are briefly introduced in Section 5.3. Following this, details of the proposed PSGR module and the segmentation framework are described in Section 5.4. Then, detailed experimental setup and all experimental results as well as the corresponding discussions are presented in Section 5.5 and Section 5.6, respectively. Finally, the paper is concluded in Section 5.7. The code for our proposed PSGR module and the segmentation framework is provided in [1].

5.2 Related Work

5.2.1 COVID-19 lesion Segmentation

A few traditional optimization-based methods have been proposed for COVID-19 lesion segmentation [152, 177, 178, 236, 244], which promote the studies of COVID-19 disease detection and treatment. For example, Qi *et al.* [152] designed an image segmentation model, named as MIS-XMACO for COVID-19 X-ray image segmentation, which introduces directional crossover strategy and directional mutation strategy to Ant colony optimization and thereby improves the segmentation performance and algorithm convergence speed. Su *et al.* [177] introduced horizontal and vertical search mechanisms to the original multi-verse optimizer to develop a multi-level thresholding method for COVID-19 chest radiography image segmentation. Meanwhile, with the success of DCNNs in medical image segmentation, various DCNNs have been also proposed for this challenging task. Xu *et al.* [217] introduced a region proposal network to a residual-inception V-Net for the segmentation of candidate infected regions in CT images. Fan *et al.* [51] developed a novel COVID-19 infection segmentation network called Inf-Net, which utilizes the reverse attention and edge-attention to improve the performance and also employs the semi-supervised learning to alleviate the shortage of high-quality annotations. Amyar *et al.* [5] proposed a multitask deep learning model to jointly identify COVID-19 patients and segment COVID-19 lesions using chest CT.

[1] `https://github.com/Haoteng0612/GraphSeg_PSGR`

This model not only leverages useful information contained in related tasks to improve both segmentation and classification, but also reduces the impact of small data problem. Qiu *et al.* [155] proposed a lightweight deep learning model called MiniSeg, which reduces the computational cost of training and can segment COVID-19 CT images efficiently. However, these models still suffer from limited segmentation performance, since none of them attempt to explore and utilize the long-range dependencies, which may overlook the rich contextual information in CT images. In this work, we incorporated GNN into a DCNN-based segmentation model to enhance the modeling of long-range dependencies and thus improve the accuracy of COVID-19 infection segmentation.

5.2.2 Global Contextual Information Learning

Constrained by the local receptive field of convolutional operations, DCNN-based segmentation models tend to have a limited ability to capture the global contextual information. To address this issue, dilated convolutions and pyramid pooling have been utilized to enlarge the receptive field of DCNNs and have showed convincing performance on semantic segmentation tasks [32–34, 242]. Recently, the non-local network [204] and PSA-Net [243] have been proposed to capture long-range dependent features, which employ the self-attention mechanism to exploit the correlations among all pixels. Meanwhile, Fu *et al.* [61] proposed a dual attention segmentation network, which creates two fully connected correlation matrices for feature and position attentions, respectively. This dual attention setting, however, may result in a significant increase of the computational cost. To reduce the computational cost of non-local methods, Huang *et al.* [91] proposed CCNet, which contains an efficient attention module called the criss-cross attention. To sum up, most self-attention methods utilize the fully connected correlation matrix to represent the feature correlations. Unfortunately, constructing a fully connected correlation matrix is not only computationally expensive but also prone to pick up noises, which may damage the semantic discriminatory power of the features. In addition, the global information learning in these models remains limited, since only low-level reasoning is performed in the image space. By contrast, the PSGR module proposed in this work performs global reasoning in the graph space in an effective way, with particular emphasis on the relation between each uncertain node and the nodes connected to it strongly.

5.2.3 Graph Reasoning for Semantic Segmentation

Many graph-based methods have been proposed for semantic image segmentation due to their superior relation reasoning capabilities. Li *et al.* [120] developed a novel approach to learning graph representations from 2D feature maps for visual recognition, which uses pixel clustering and feature similarity measurement to transform an image to a graph structure. Chen *et al.* [36] performed relational reasoning by projecting a set of features that are globally aggregated over the coordinate space into an interaction space. Graph reasoning has also been applied to medical image segmentation. Soberanis-Mukul *et al.* [169] combined uncertainty analysis and graph convolutional network (GCN) to refine organ segmentation in CT images. Liu *et al.* [125] utilized a predefined pseudo landmark to project mammogram images to the graph space and then introduced a bipartite GCN to endow DCNN segmentation networks with the cross-view reasoning ability. However, the feature mapping strategies used in these methods rely highly on either the prior knowledge or a predetermined number of nodes, which tends to result in limited generalizability and adaptiveness. Hu *et al.* [84] constructed the graph in a pixel-wise and class-wise manner and performed graph reasoning on dynamically sampled pixels, which avoids all those predetermined and inflexible feature projections and exploits contextual information for semantic segmentation. However, the connections are only restricted among those sampled pixels, which may lead to an insufficient aggregation of effective information. Li *et al.* [117] constructed the fully connected graph in a pixel-wise way and organized the graph reasoning as a spatial pyramid. However, similar to those self-attention methods, the semantic discriminatory of the features may be ignored when the feature maps are represented by fully connected graphs. By contrast, our PSGR module constructs a sparse graph from the perspective of the message passing mechanism of GNN, where each node can selectively connect to the nodes from which it can gain more effective information. This design can facilitate GNN to capture the long-range information in the graph reasoning stage.

5.3 Preliminaries

An attributed and weighted graph G with N nodes is denoted by (A, H), where $A \in \mathcal{R}^{N \times N}$ is the graph adjacency matrix, $H \in \mathcal{R}^{N \times c}$ is the node feature matrix, and c is the dimensionality of the feature at each node. The node latent feature matrix Z, which represents the embedded node features in the latent space $\mathcal{Z}$, can be formally expressed as follows

$$Z^{(k)} = F(A^{(k-1)}, Z^{(k-1)}; \theta^{(k)}), \tag{5-29}$$

where k denotes the $k-th$ layer of GNN, $A^{(k-1)}$ is the graph adjacency matrix computed by the $(k-1)-th$ layer of the GNN, $\theta^{(k)}$ is the ensemble of trainable parameters in the $k-th$ layer, and $F(\cdot)$ is the forward function to aggregate and transform the messages across the nodes. Particularly, $Z^0 = H$. Many previous studies specified different definitions of function $F(\cdot)$ [65, 71] such as the graph convolution neural network (GCN) [102] and higher-order GCN (HO-GCN) [144]. The GCN combines the information of the neighborhoods as the node representation linearly. The HO-GCN takes higher-order graph structures into account, which is important to capture the long-range information in the graph.

5.4 Methods

The proposed COVID-19 infection segmentation model consists of a segmentation backbone, a coarse segmentation branch, and a novel PSGR module that is inserted between the encoder and decoder of the backbone. The diagram of this model is illustrated in Figure 15. We now delve into its details.

5.4.1 PSGR Module

The PSGR module aims to improve the effectiveness of information gain on uncertainly segmented pixels and hinder the noise propagation in long-range information reasoning, and therefore it can further boost the segmentation performance especially in those uncertainly

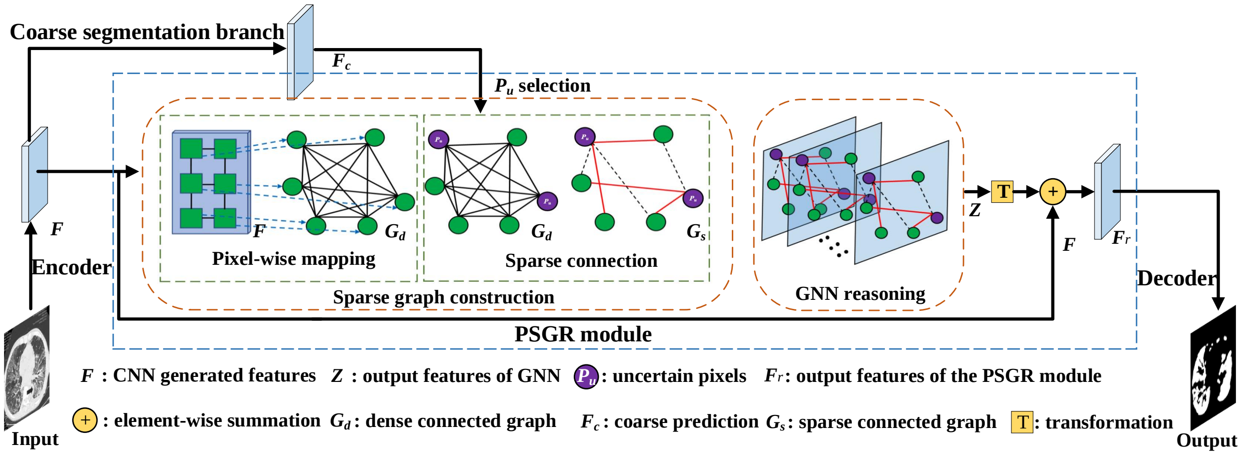

Coarse segmentation branch
F_c
P_u selection
F
Encoder
F
G_d
Pixel-wise mapping
G_d
G_s
Sparse connection
Sparse graph construction
GNN reasoning
Z
T
F
F_r
Decoder
PSGR module
Input
Output
F : CNN generated features
Z : output features of GNN
P_u: uncertain pixels
Fr: output features of the PSGR module
+ : element-wise summation
G_d: dense connected graph
F_c: coarse prediction
G_s: sparse connected graph
T: transformation

Figure 15: Diagram of the proposed segmentation model, including a segmentation backbone, a coarse segmentation branch, and the proposed PSGR module. The input is the CT image, which has been pre-processed with zero mean and unit variance intensity normalization.

segmented regions. The PSGR module is composed of two components: sparse graph construction and long-range information reasoning (see Figure 15).

5.4.1.1 Sparse Graph Construction

Let the feature map generated by the encoder be denoted by $X \in \mathcal{R}^{h\times w\times c}$, where $h \times w$ is the feature size and c is the dimensionality. The node feature matrix H can be obtained by reshaping X to the size of $N \times c$, where $N = h \times w$. After mapping each pixel to a graph node, the constructed graph $G = (A, H)$ preserves the inherent information of each pixel and can provide precise pixel-wise information for global reasoning. The adjacency matrix A encodes the connections (edges) between the nodes. Since it is neither necessary nor computationally tractable to fully connect all nodes, we construct a sparsely-connected graph G_s based on the information theory.

Connectivity Distribution Matrix. Suppose two pixels p_i and p_j are mapped to two nodes v_i and v_j, respectively. The connection between v_i and v_j is measured by the inner product of the features of p_i and p_j. Thus, a higher similarity between p_i and p_j indicates a stronger connection between v_i and v_j. The feature similarity matrix S is defined as follows

$$S = HH^T - HH^T \odot I, \tag{5-30}$$

where I is the identity matrix, $\odot$ is element-wise product, and the second term is used to ensure that the diagonal elements of S are zero. The feature similarity matrix S can be regarded as the adjacency matrix of a densely-connected graph G_d (see Figure 15). Then, we construct the normalized node connectivity distribution matrix $\hat{S}$ by computing the graph Laplacian

$$\hat{S} = D^{-\frac{1}{2}} S D^{-\frac{1}{2}}, \tag{5-31}$$

where D is the degree matrix of S. Note that the i-th line in $\hat{S}$, denoted by $\hat{S}_{i:}$, representing the connectivity probability distribution between v_i and any other nodes and $\sum \hat{S}_{i:} = 1$.

Node Information Score. For each node v_i, we define an information score (IS) to measure the information quantity that v_i gains from each of its neighbors, shown as follows

$$IS_i = \left\| \hat{S}_{i:}^T \otimes H \right\|_{\tilde{L}_1}, \tag{5-32}$$

where $\|\cdot\|_{\tilde{L}_1}$ is line-wise L_1 norm, and $\otimes$ is the scalar-multiplication between each line of two matrices. Particularly, $\hat{S}_{i:}^T$ represents the connectivity between node i to all other nodes in graph. The computation of the information score (see Equation 5-32) of v_i can be derived as

$$IS_i = \sum_{j=1}^{N} \left\| \hat{S}_{i,j} \cdot H_j \right\|_{L_1}, \tag{5-33}$$

where $\hat{S}_{i,j}$ is the information propagation rate between v_i and v_j. $(\hat{S}_{i,j} \cdot H_j)$ is the information that v_i gains from v_j. The L_1 norm is utilized to yield the information gain that the v_i obtains from each node v_j.

Sparse Connection Adjacency Matrix. The key to construct a sparsely-connected graph is the criterion that can guide edge pruning. We divide all nodes into certain nodes and uncertain nodes and then define the criterion as: (1) the connection between any pair of certain nodes is removed, and (2) for each uncertain node, only the connections between it and K neighbors with highest IS values are preserved. The certainty of each node is determined based on the predictions made by the coarse segmentation branch. Specifically, for each node, we calculate the difference between the largest and second largest predicted probabilities of the corresponding pixel belonging to a region. Then, we select $R_u \times N$ nodes with lowest probability difference as uncertain nodes, where R_u is the uncertain nodes selection ratio. As a result, each element of the sparse connection adjacency matrix ($\widetilde{A}$) can be formally expressed as:

$$\widetilde{A}_{ij} = \{\hat{S}_{ij} \mid v_i \in \Omega_u, v_j \in topK\lceil IS_i \rceil\}, \tag{5-34}$$

where Ω_u is the set of uncertain nodes, $topK\lceil IS_i \rceil$ generates a set containing K neighbors of v_i with highest *IS* values, and K is empirically set to $N/2$ for this study. Then, we obtain a sparsely-connected graph $G_s = (\widetilde{A}, H)$.

5.4.1.2 Long-Range Information Reasoning with HO-GNN

The higher-order information, which is aggregated from global neighbors via multi-hops, is difficult to capture but important in reasoning the contextual relations in the graph. Since HO-GNN [144] is a powerful tool to capture both local and global information in graph-structured data, we utilize HO-GNN to perform graph reasoning in our PSGR module.

The way that HO-GNN aggregates and propagates the information can be formulated as:

$$Z^{(k)}(v_i) = \mathcal{F}(Z^{(k-1)}(v_i)\theta_1^{(k-1)} + \sum_{v_j \in \Phi(v_i)} Z^{(k-1)}(v_j)\theta_2^{(k-1)}), \tag{5-35}$$

where $\Phi(v_i) = N_l(v_i) \cup N_g(v_i)$ is the union of the local and global neighborhoods of node v_i, $Z(v_i)$ is the latent feature of v_i, $\mathcal{F}(\cdot)$ is a nonlinear transformation function ($e.g.$, sigmoid), θ_1 and θ_2 are trainable parameters, and k is the index of layers. In the stage of graph reasoning, each uncertain node can aggregate information from its local and global neighborhoods, enabling the retrieval of long-range contextual dependencies.

Once obtaining the feature map produced by HO-GNN ($i.e.$, Z), we first reshape it back to the size of $h \times w \times c$, and then fuse it with the input feature map F via element-wise summation to generate the output feature map of the PSGR module, denoted by F_r.

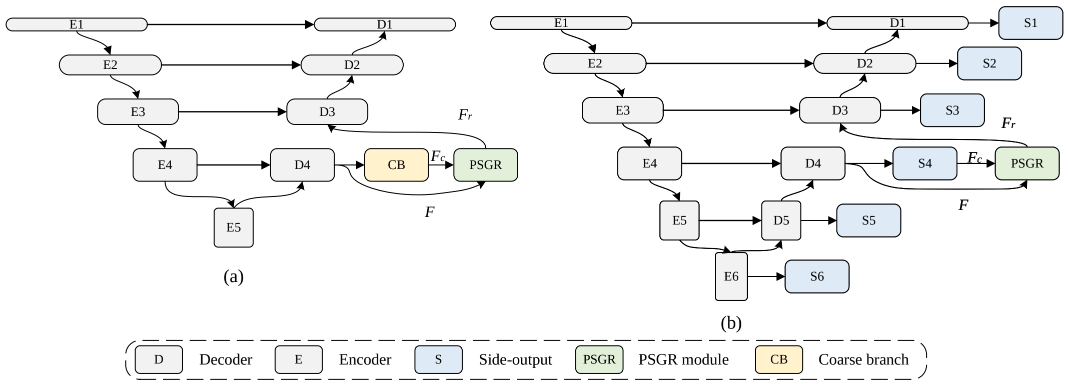


Figure 16: Deploying the coarse segmentation branch and proposed PSGR module in the segmentation backbone. (a) and (b) represent U-Net and U^2-Net, respectively. See 5.4.2 for details.

5.4.2 Segmentation Model with PSGR Module

Segmentation Backbone. For this study, we choose two widely-used baselines as the segmentation backbones, *i.e.*, U-Net [158] and U^2-Net [154]. The former has shown convincing and robust performance on a large variety of medical image segmentation tasks, and the latter has special two-level nested U-structure which can help to capture abundant contextual information and thereby obtained superior performance on several computer vision tasks. Here we adopt all default configurations used in the official implementations[2],[3], except for replacing the transposed convolution with the bi-linear interpolation in U-Net.

Coarse Segmentation Branch. To determine uncertain nodes, we need a coarse segmentation branch to predict the rough probability of a pixel belonging to each region. Since the features in deep stages may have too low a spatial resolution to recover the details, we place the coarse segmentation branch after the fourth stage of the decoder in the backbone (see Figure 16), where the feature map has 1/8 size of the input image. For U-Net, we first apply a layer sequence of $3{\times}3\boldsymbol{Conv}+\boldsymbol{BN}+\boldsymbol{ReLU}+1{\times}1\boldsymbol{Conv}$ to produce the coarse prediction map $\boldsymbol{F_c}$ where the middle channel is set to 128. Besides feeding $\boldsymbol{F_c}$ to the PSGR module, we

[2] `https://github.com/milesial/Pytorch-UNet`

[3] `https://github.com/xuebinqin/U-2-Net`

also upsample it to the input size as an auxiliary deep supervision. Considering U^2-Net has side-output for each stage of the decoder, we directly adopt its fourth side-output as F_c.

Deploying PSGR Module. As illustrated in Figure 15 and Figure 16, the PSGR module takes both F and F_c as inputs, and directly produces refined feature map F_r, which is enhanced with global long-range dependencies. Inside our PSGR module, we specially apply 1×1 convolutions to keep the size and channel number of F and F_r consistent. Due to its pixel-wise mapping strategy and flexible adaptability, our PSGR module can also be easily incorporated into any other segmentation networks in an end-to-end-training fashion.

Loss Function and Supervision Manner. Since we adopt the coarse segmentation result s_{coarse} for auxiliary supervision, the loss function is defined as follows

$$L = L_{seg}(s_{main}, y) + \lambda L_{seg}(s_{coarse}, y), \tag{5-36}$$

where s_{main} is the segmentation results produced by the backbone, y is the ground truth, and the weighting parameter λ is set to 0.5 for all experiments without further tuning. Each segmentation loss L_{seg} is the sum of the binary cross-entropy (BCE) loss and Dice loss, shown as follows

$$L_{seg} = \ell_{BCE} + \ell_{Dice}. \tag{5-37}$$

5.5 Experimental Setup

5.5.1 Datasets

The COVID-19 CT segmentation (COVID19-CT-100) dataset [136], COVID-19 CT lung and infection segmentation (COVID19-CT-Seg20) dataset [95], and MosMedData [142] were used for this study. COVID19-CT-100 was collected by the Italian Society of Medical and Interventional Radiology[4]. It consists of 100 COVID-19 infected CT slices from >40 patients. Since the annotations of different infected regions (ground-glass opacity (GGO) and consolidation) were provided, we follow [155] and [51] to evaluate the segmentation performance of our method on bi-class segmentation and multi-class segmentation, respectively.

[4] `https://sirm.org/category/senza-categoria/covid-19/`

COVID19-CT-Seg20 contains 20 COVID-19 CT images, where lungs and infections were annotated by two radiologists and verified by an experienced radiologist. Here we only focused on the segmentation of the COVID-19 infection, since it is more challenging and important. MosMedData was collected by the Research and Practical Clinical Center for Diagnostics and Telemedicine Technologies of the Moscow Health Care Department. A total of 50 CT scans, each having less than 25% lung infected, were selected and manually labeled by experts. Since all the infected regions were annotated with one class label, we only conduct bi-class segmentation on COVID-19-CT-Seg20 and MosMedData datasets. Considering the limited scans and large inter-slice spacing of those volumetric data, we followed previous work [51, 155] to perform 2D segmentation on all datasets. As a result, we totally have 100, 1844, and 785 2D CT slices from COVID19-CT-100, COVID19-CT-Seg20, and MosMedData, respectively. In this study, the effects of subjects' age, gender and race or any other variables on the results are not evaluated since the related information is not released by the provider. The details of these datasets were shown in Table 14.

5.5.2 Implementation Details

In the training phase, we first applied data augmentation techniques on the fly to reduce potential overfitting, including random scaling (0.8 to 1.2), random rotation (±15°), random intensity shift of (±0.1) and intensity scaling of (0.9 to 1.1). Then, we cropped or padded each image to a size of 512×512. The training iterations were set to 200 epochs with a linear warmup of the first 5 epochs. We trained the model using the Adam optimizer with a batch size of 8 and synchronized batch normalization. The initial learning rate was set to $1e^{-3}$ and decayed by $(1 - \frac{current\_epoch}{max\_epoch})^{0.9}$. We also regularized the training with an l_2 weight decay of $1e^{-5}$. The uncertain pixel selection ratio R_u was set to 0.005, 0.01, and 0.005 on COVID19-CT-100, COVID19-CT-Seg20, and MosMedData, respectively. In the inference phase, we only applied padding operations to the input image if its size can not be divisible by the down-sample rate of the model. We used five-fold cross-validations for bi-class segmentation where the data division in [155][5] was adopted. As for multi-class segmentation, we adopted

[5] `https://github.com/yun-liu/MiniSeg`

Table 14: Details of three public datasets.

Datasets	Slice number	Resolution
COVID19-CT-100 [136]	100	512×512
COVID19-CT-Seg20 [95]	1844	512×512-630×630
MosMedData [142]	785	512×512

the data division in [51][6] and divided the COVID19-CT-100 dataset into a training set, a validation set, and a test set. It is notable that we performed the experiments independently on three datasets, *i.e.*, not combining the data from different datasets for training. All experiments were conducted based on PyTorch 1.6.0 and PyTorch Geometric [53] 1.6.1 and were deployed on a workstation with 4 NVIDIA TITAN XP GPUs.

5.5.3 Baselines and Evaluation Metrics

Besides evaluating the effectiveness of our PSGR module against U-Net and U^2-Net in the ablation study, we also compared our approach with eight segmentation baselines, *i.e.*, FCN-8s [126], DeepLabv3+ [34], U-Net++ [247], Attention U-Net [146], Inf-Net [51], MiniSeg [155], DANet [61], and CCNet [91]. U-Net++ and Attention U-Net are two well-performing baselines in medical image segmentation, while FCN-8s and DeepLabv3+ are two popular baselines in semantic segmentation. MiniSeg and Inf-Net are two state-of-the-art (SOTA) models which have shown convincing performance in COVID-19 segmentation. DANet and CCNet are introduced as two cutting-edge attention-based networks which also focus on enhancing long-range dependencies for semantic segmentation models.

We adopted six metrics to assess the performance of segmentation models, including the mean intersection over union (mIoU), Dice similarity coefficient (DSC), sensitivity (SEN), specificity (SPE), Hausdorff distance (HD), and mean absolute error (MAE). Specifically, mIoU, DSC, SEN, and SPE are four overlap-based metrics, each ranging from 0 to 1 and a larger value indicating better performance. HD is a shape distance-based metric, which can

[6] `https://github.com/DengPingFan/Inf-Net`

be used to measure the dissimilarity between the surfaces / boundaries of the segmentation result and the ground-truth. MAE can represent the dissimilarity between the segmentation result and the ground-truth. As for HD and MAE, a lower value indicates a better segmentation result. We followed [155] to adopt mIoU, SEN, SPE, DSC, and HD for bi-class infection segmentation tasks, and followed [51] to choose DSC, SEN, SPE, and MAE for the multi-class infection segmentation task.

5.6 Results and Discussions

5.6.1 Comparative Experiments

Performance in Bi-class Infection Segmentation. Table 15 gives the performance of our models and eight competing ones, including FCN-8s [126], DeepLa-bv3+ [34], U-Net++ [247], Attention U-Net [146], DANet [61], CCNet [91], Inf-Net [51], and MiniSeg [155] in bi-class infection segmentation on the COVID19-CT-100, COVID19-CT-Seg20, and MosMedData datasets. It shows that our models (*i.e.*, U-Net equipped with our PSGR module (U-Net+PSGR) and U^2-Net equipped with our PSGR module (U^2-Net+PSGR)) outperform all competing methods substantially and consistently in terms of DSC and mIoU, indicating that the segmentation results of our models match well with the ground-truth. Across all metrics, U^2-Net+PSGR and U-Net+PSGR achieve the overall best and second best performance, respectively. Meanwhile, comparing to two self-attention based SOTAs, *i.e.*, DANet [61] and CCNet [91], our models achieve clearly superior segmentation results, which tend to show the strong ability in capturing long-range dependencies for COVID-19 infection segmentation. At last, it is remarkable that using our PSGR module can significantly reduce the HD values when comparing to any competing models, which demonstrates that the boundaries detected in our segmentation results match the ground-truth boundaries very well.

In addition, the bi-class infection segmentation results produced by U-Net, U^2-Net, MiniSeg, and our models, and the corresponding ground-truths were visualized in Figure 17.

It shows that, compared to three competing models, our U-Net+PSGR and U^2-Net+PSGR can generate the infectious regions that match better with the ground-truths, especially when those regions are disperse and tiny. Comparing the results of U-Net+PSGR and U-Net (or U^2-Net+PSGR and U^2-Net), we can conclude that the improvements of segmentation performance should be attributed to the strong long-range information reasoning ability of our PSGR module.

Performance in Multi-class Infection Segmentation. Table 16 gives the performance of our models and five competing ones, including FCN-8s [126], U-Net [158], DeepLabv3+ [34], and Inf-Net [51] (with two backbones), in multi-class infection segmentation on the COVID-19-CT-100 dataset. It reveals that Inf-Net (FCN-8s) [51] has better segmentation performance than DeepLabv3+ [34], FCN-8s [126], and U-net [158], and achieve best DSC and SEN on the GGO segmentation task. Our U-Net+PSGR and U^2-Net+PSGR achieve best performance across all metrics in the segmentation of consolidation, which is more challenging since each consolidation region tends to have a tiny size. In addition, our models have consistently and significantly lower MAE than other models on both GGO and consolidation segmentation, which indicates again that our segmentation results have less mismatched predictions. In summary, both U-Net+PSGR and U^2-Net+PSGR achieve overall best performance across all metrics. It is worth noting that, different from Inf-Net, which actually performs semi-supervised segmentation using 1600 extra unlabelled CT slices, we only use 50 CT slices from the COVID19-CT-100 dataset to train our model for this challenging segmentation task.

We also visualized the qualitative segmentation results in Figure 18. It reveals that the results produced by our U-Net+PSGR and U^2-Net+PSGR are much more similar to the ground-truths than those generated by DeepLabv3+ and FCN-8s, and are comparable to the results of Inf-Net(U-Net). Note that Inf-Net was trained with a huge number of external data.

All these convincing results on three datasets for both bi-class and multi-class segmentation tasks demonstrate the effectiveness and strong generalizability of our U-Net+PSGR and U^2-Net+PSGR models.

5.6.2 Ablation Study

We conducted an ablation study on all three public datasets (i.e., COVID19-CT-100, COVID19-CT-Seg20, and MosMedData) under a bi-class segmentation setting to evaluate the effectiveness of our PSGR module. We compared our U-Net+PSGR and U^2-Net+PSGR to their baselines U-Net [158] and U^2-Net [154], respectively. Besides, since U-Net has no deep supervision structure, we added a coarse segmentation branch (CSB) to U-Net to provide deep supervision and reported the results, too. The results in Table 17 show that solely introducing CSB to U-Net can improve DSC from 68.37% to 76.88%, 63.60% to 73.64%, and 59.04% to 61.89% on COVID19-CT-100, COVID19-CT-Seg20, and MosMedData, respectively. In addition, we state that integrating our PSGR module to U-Net or U^2-Net can substantially improve the segmentation accuracy and achieve the best performance in terms of all metrics on all datasets. The consistent performance gains over baselines demonstrate the effectiveness of our PSGR module for COVID-19 infection segmentation.

5.6.3 Visualization of the PSGR Module

To further demonstrate the ability of our PSGR module to capture long-range dependencies, we also provide some visualization results in Figure 19. Concretely, we chose six images (two from each dataset) as a case study where the CT images and ground-truths are provided in the first row of Figure 19.

Then, we randomly selected a pixel of an infected region on each image, and visualized the corresponding row in the sparse connection adjacency matrix $\widetilde{A}$ in the second row. It reveals that our PSGR module can accurately capture long-range dependencies with respect to specific semantic information. For instance, in the image of the second column, the infected region in the green box is quite difficult to segment since it is tiny and isolated. Fortunately, given a pixel in this region (marked as a red dot), our PSGR module can successfully highlight other foreground pixels (highlighted in white) from the global, where the useful contextual information exists, to facilitate the segmentation task.

5.6.4 Parameter Analysis

We analyze the impact of three different hyperparameters on our segmentation performance in this section. In the proposed PSGR module, the hyperparameter $\boldsymbol{R_u}$ represents the ratio of how many uncertain nodes should be selected, and the parameter $\boldsymbol{K}$ ratio represents the ratio of how many neighbor nodes are connected to the selected uncertain nodes. To investigate the impact of these two parameters on the segmentation performance, we plotted the DSC and HD values obtained on the COVID19-CT-100 dataset versus the values of $\boldsymbol{R_u}$ and $\boldsymbol{K}$ ratio in Figure 20 and Figure 21, respectively. Since infectious regions occupy around 1% area on most COVID-19 CT slices, we increased the value of $\boldsymbol{R_u}$ from 0 to 0.02 with a step of 0.005. Ideally, the $\boldsymbol{K}$ ratio can be set up in the range of $[0, 1]$ so that we increased the value of $\boldsymbol{K}$ ratio from 0 to 1 with a step of 0.2. It shows that, with the increase of $\boldsymbol{R_u}$ and $\boldsymbol{K}$ ratio, the segmentation performance of U-Net+PSGR and U^2-Net+PSGR tends to incline and then decline. It indicates that large $\boldsymbol{R_u}$ and $\boldsymbol{K}$ ratios degrade the segmentation performance, which may be attributed to the redundant information and noise introduced by excessive node connections during the graph reasoning process. The best values of $\boldsymbol{R_u}$ and $\boldsymbol{K}$ ratio for both models, which lead to the highest DSC and lowest HD, are 0.005 and 0.5, respectively. In addition, Figure 20 and Figure 21 also indicate that using our PSGR module with different parameter values consistently outperforms the baselines (see blue dash-lines in the figures), which again justifies the robustness and effectiveness of the proposed PSGR module. Beyond these two parameters, we also analyze the impact of loss weight (i.e., λ) on the model segmentation performance in Figure 22. We change the λ parameter from 0 to 1 with a step of 0.2. It indicates that the performance of the whole framework is consistent with the increase of λ values and the optimal λ value is around 0.5 for both U-Net+PSGR and U^2-Net+PSGR. It also shows that the whole framework can yield better DSC and HD results than the best baseline result (i.e., see green dashlines in the figure presenting the DSC and HD results obtained from MiniSeg) when using all different loss weights.

5.7 Conclusion

In this paper, we propose an effective graph reasoning module called PSGR to capture long-range contextual information and we incorporate it into different segmentation backbones to improve the segmentation of COVID-19 infection in CT images. The PSGR module has two advantages over existing graph reasoning techniques for semantic segmentation. First, the pixel-wise mapping strategy used for convert an image into a graph not only avoids imprecise pixel-to-node projections but also preserves the inherent information of each pixel. Second, the edge pruning method used to construct a sparsely-connected graph results in effective information retrieval and reduces the noise propagation in GNN-based graph reasoning. Our results show that the segmentation networks equipped with our PSGR module outperform several widely-used segmentation models on three public datasets. Several directions might be considered as our future works. First, we plan to reduce the computation cost of our proposed PSGR module, which may facilitate an extension of our module to $3\boldsymbol{D}$ medical image segmentation. Moreover, though outperforming to baseline methods consistently, our PSGR module relies on determining the best choices of two hyperparameters (i.e., $\boldsymbol{R_u}$ and $\boldsymbol{K}$ ratio). Therefore, a parameter adaptive (or parameter-free) conception is valuable to be introduced to our PSGR module.

Table 15: Quantitative results of different methods on three public datasets. The best and second best results are shown in red and blue, respectively.

Methods	COVID19-CT-100					COVID19-CT-Seg20					MosMedData				
	mIoU	SEN	SPE	DSC	HD	mIoU	SEN	SPE	DSC	HD	mIoU	SEN	SPE	DSC	HD
FCN-8s [126]	71.85	66.47	93.56	58.11	104.68	82.54	84.10	98.02	73.60	51.47	70.51	80.75	97.08	53.33	84.43
DeepLabv3+ [34]	79.45	79.58	97.55	71.70	93.09	81.26	81.61	95.35	42.79	182.14	74.14	74.65	97.26	57.16	102.78
U-Net++ [247]	77.64	77.26	97.28	69.04	91.73	80.73	79.61	96.75	70.34	63.01	73.39	75.67	96.13	59.08	88.21
Attention U-Net [146]	77.71	74.75	97.56	68.93	92.15	80.70	82.92	97.41	71.27	64.91	74.62	81.32	97.63	59.34	95.16
DANet [61]	73.57	66.30	92.76	61.34	99.11	81.59	88.78	99.13	73.82	114.69	73.47	75.00	95.80	56.07	74.04
CCNet [91]	75.24	69.55	95.92	63.99	98.03	81.27	86.61	99.16	73.93	90.84	72.02	79.16	96.29	54.83	83.07
Inf-Net [51]	81.62	76.50	98.32	74.44	86.81	64.62	69.46	99.02	63.38	79.68	74.32	62.93	93.45	56.39	71.77
MiniSeg [155]	82.15	84.95	97.72	75.91	74.42	84.49	85.06	99.05	76.27	51.06	78.33	79.62	97.71	64.84	71.69
U-Net+PSGR	86.58	83.62	98.86	83.16	50.68	87.88	77.83	99.78	78.58	46.86	80.16	72.73	99.86	66.95	68.49
U^2-Net+PSGR	87.92	85.89	98.95	84.83	42.85	87.55	79.78	99.75	78.32	43.51	80.52	72.30	99.88	67.27	64.45

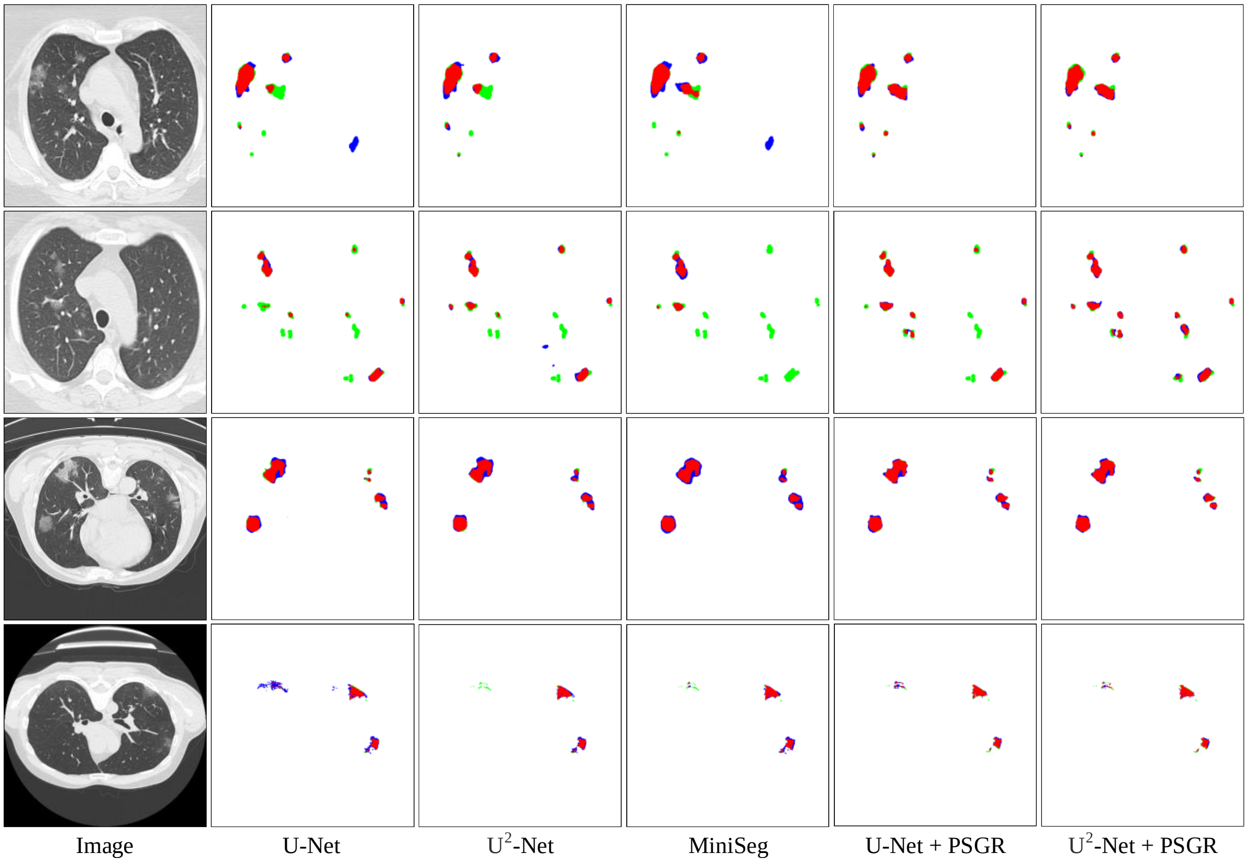


Figure 17: Visualization of the bi-class infection segmentation results produced by our models and three competing ones on the COVID19-CT-100 (row 1 and 2), COVID19-CT-Seg20 (row 3), and MosMedData (row 4) datasets. The true positive, false negative, and false positive are highlighted with red, green, and blue, respectively. Comparing the results in column 2 and 4 (or column 3 and 5), we can conclude that the segmentation improvements should be attributed to the strong long-range information reasoning ability of our PSGR module. Better view with colors and zooming in.

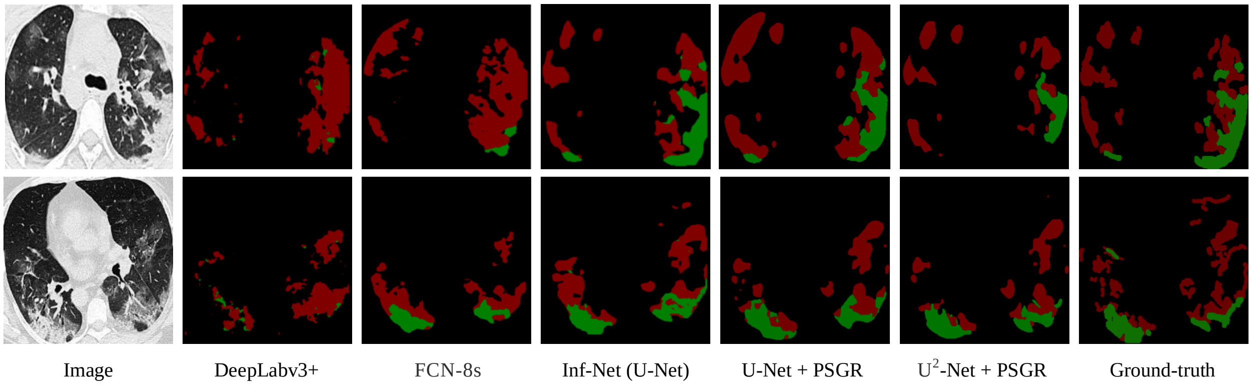


Figure 18: Visualization of the multi-class infection segmentation results on the COVID19-CT-100 dataset produced by our models and three competing ones. The regions of GGO and Consolidation are highlighted in red and green, respectively. Better view with colors and zooming in.

Table 16: Quantitative results of different methods for multi-class infection segmentation on the COVID119-CT-100 dataset. The best and second best results are shown in red and blue, respectively. The values of DSC, SEN, SPE, and MAE are in percentage terms. $*$ represents using extra training data.

Methods	GGO				Consolidation				Average			
	DSC	SEN	SPE	MAE	DSC	SEN	SPE	MAE	DSC	SEN	SPE	MAE
DeepLabv3+ [34]	44.3	71.3	82.3	15.6	23.8	31.0	70.8	7.7	34.1	51.2	76.6	11.7
FCN-8s [126]	47.1	53.7	90.5	10.1	27.9	26.8	71.6	5.0	37.5	40.3	81.1	7.6
U-Net [158]	44.1	34.3	98.4	8.2	40.3	41.4	96.7	5.5	42.2	37.9	97.6	6.6
Inf-Net (FCN-8s)* [51]	64.6	72.0	94.1	7.1	30.1	23.5	80.8	4.5	47.4	47.8	87.5	5.8
Inf-Net (U-Net)* [51]	62.4	61.8	96.6	6.7	45.8	50.9	96.7	4.7	54.1	56.4	96.7	5.7
U-Net+PSGR	62.3	69.3	97.9	3.2	49.0	65.3	98.4	2.1	55.7	67.3	98.2	2.7
U^2-Net+PSGR	60.2	58.5	98.9	2.9	49.8	50.1	99.2	1.6	55.0	54.3	99.1	2.3

Table 17: Ablation studies of our proposed PSGR module on three public datasets. The best results are shown in red.

Methods	COVID19-CT-100					COVID19-CT-Seg20					MosMedData				
	mIOU	SEN	SPE	DSC	HD	mIOU	SEN	SPE	DSC	HD	mIOU	SEN	SPE	DSC	HD
(a) U-Net	77.56	72.24	97.71	68.37	94.25	78.72	68.33	97.61	63.60	87.14	71.89	62.88	96.99	59.04	107.37
(b) U-Net + CSB	82.01	76.81	98.58	76.88	71.01	83.24	71.49	99.46	73.64	65.65	75.92	66.80	99.57	61.89	95.59
(c) U-Net + CSB + PSGR	86.58	83.62	98.86	83.16	50.68	87.88	77.83	99.78	78.58	46.86	80.16	72.73	99.86	66.95	68.49
(d) U^2-Net	80.46	76.92	97.62	75.87	75.78	80.12	71.44	98.40	70.29	77.07	73.71	64.75	98.53	60.17	99.11
(e) U^2-Net + PSGR	87.92	85.89	98.95	84.83	42.85	87.55	79.78	99.75	78.32	43.51	80.52	72.30	99.88	67.27	64.45

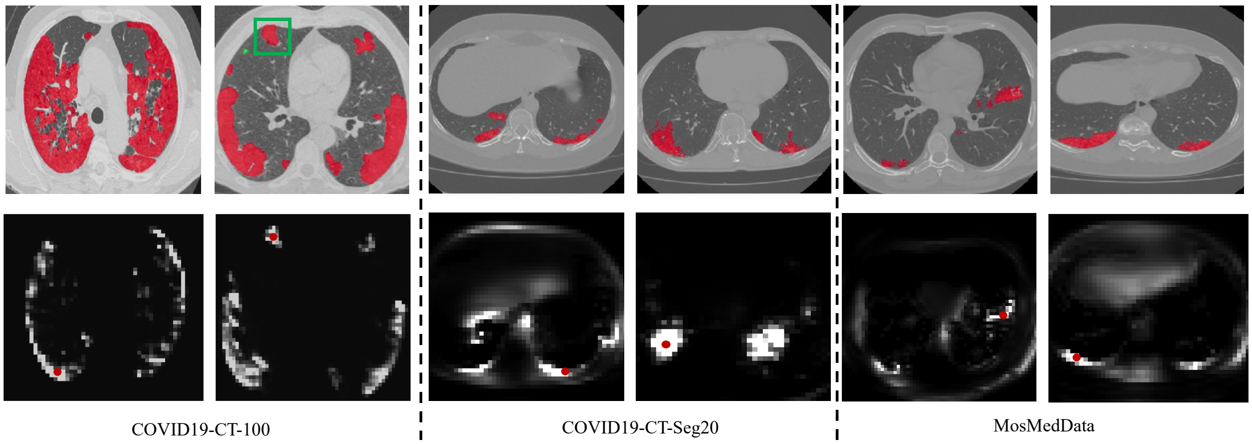


Figure 19: Visualization of the ability of our PSGR module to capture long-range dependencies on three datasets. Given a pixel (red dot) in an infected region, our PSGR module can highlight other foreground pixels (highlighted in white) from the entire image, where the contextual information exists.

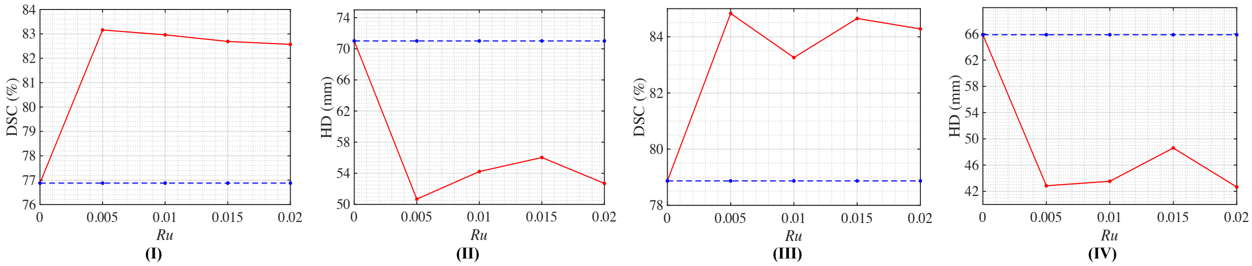


Figure 20: Impact of R_u on segmentation performance: (I) DSC of U-Net+PSGR v.s. R_u, (II) HD of U-Net+PSGR v.s. R_u, (III) DSC of U^2-Net+PSGR v.s. R_u, and (IV) HD of U^2-Net+PSGR v.s. R_u. The blue dashlines are the baseline results where the proposed PSGR module is not integrated. Higher DSC values or lower HD values indicates better segmentation performance.

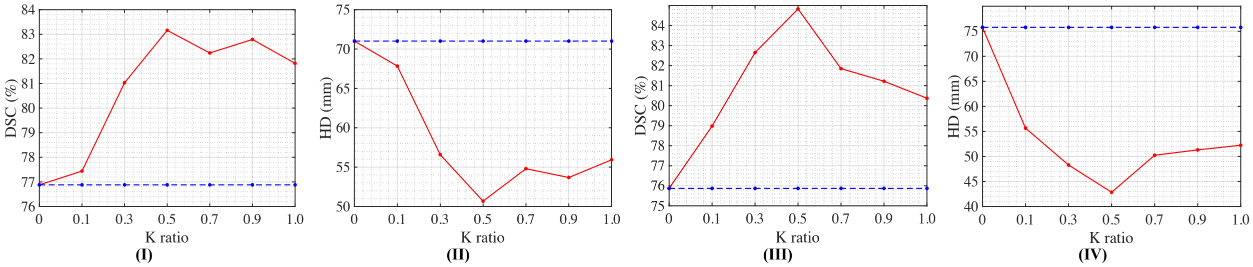


Figure 21: Impact of K ratio on segmentation performance: (I) DSC of U-Net+PSGR v.s. K ratio, (II) HD of U-Net+PSGR v.s. K ratio, (III) DSC of U^2-Net+PSGR v.s. K ratio, and (IV) HD of U^2-Net+PSGR v.s. K ratio. The K ratio is computed by: $K/$ # of nodes in the constructed graph. The blue dashlines are the baseline results where the proposed PSGR module is not integrated. Higher DSC values or lower HD values indicates better segmentation performance.

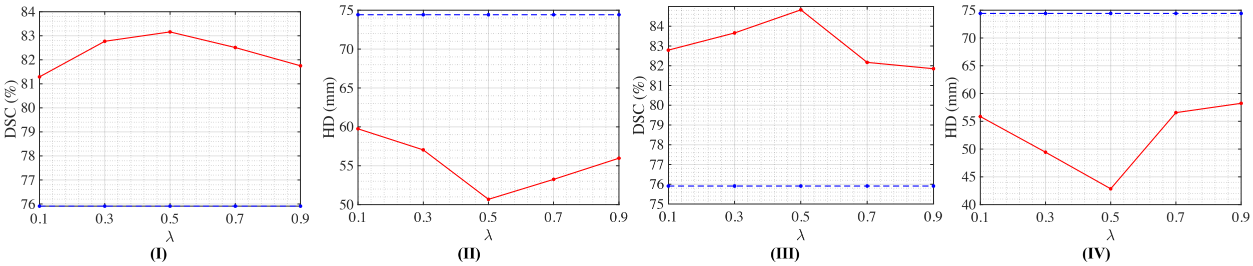


Figure 22: Impact of λ on segmentation performance: (I) DSC of U-Net+PSGR v.s. λ, (II) HD of U-Net+PSGR v.s. λ, (III) DSC of U^2-Net+PSGR v.s. λ, and (IV) HD of U^2-Net+PSGR v.s. λ. The blue dashlines are the baseline results yielded from MiniSeg. Higher DSC values or lower HD values indicate better segmentation performance.

6.0 Conclusion

Several interpretable graph learning methods proposed in my Ph.D. studies are introduced in this thesis. In particular, a hierarchical graph pooling model and a hierarchical signed graph pooling model are proposed for unsigned and signed graph representation learning, respectively. These models are applied on multumodal brain networks to predict the brain diseases and clinical phenotypes. The experimental results show that our proposed models can outperform state-of-the-arts baseline models in brain network predictions, and beyond, they can identify significant brain biomarkers that related to the prediction targets. Another application study is also presented in this thesis, where we propose a pixel-wise graph reasoning (PSGR) module to capture the long-range dependencies in the process of COVID-19 CT image segmentation. The experimental results demonstrate the superiority of our method comparing to different baseline segmentation methods, and the contextual information captured by our module is also visualized in our study.

Bibliography

[1] Bryon Adinoff, Mark J Williams, Susan E Best, Thomas S Harris, Patricia Chandler, and Michael D Devous Sr. Sex differences in medial and lateral orbitofrontal cortex hypoperfusion in cocaine-dependent men and women. *Gender medicine*, 3(3):206–222, 2006.

[2] Tao Ai, Zhenlu Yang, Hongyan Hou, Chenao Zhan, Chong Chen, Wenzhi Lv, Qian Tao, Ziyong Sun, and Liming Xia. Correlation of chest ct and rt-pcr testing for coronavirus disease 2019 (covid-19) in china: a report of 1014 cases. *Radiology*, 296(2):E32–E40, 2020.

[3] Olusola Ajilore, Liang Zhan, Johnson GadElkarim, Aifeng Zhang, Jamie D Feusner, Shaolin Yang, Paul M Thompson, Anand Kumar, and Alex Leow. Constructing the resting state structural connectome. *Frontiers in neuroinformatics*, 7:30, 2013.

[4] Nicola Amoroso, Marianna La Rocca, Stefania Bruno, Tommaso Maggipinto, Alfonso Monaco, Roberto Bellotti, and Sabina Tangaro. Brain structural connectivity atrophy in alzheimer's disease. *arXiv preprint arXiv:1709.02369*, 2017.

[5] Amine Amyar, Romain Modzelewski, Hua Li, and Su Ruan. Multi-task deep learning based ct imaging analysis for covid-19 pneumonia: Classification and segmentation. *Computers in Biology and Medicine*, 126:104037, 2020.

[6] Salim Arslan, Sofia Ira Ktena, Ben Glocker, and Daniel Rueckert. Graph saliency maps through spectral convolutional networks: Application to sex classification with brain connectivity. In *Graphs in biomedical image analysis and integrating medical imaging and non-imaging modalities*, pages 3–13. Springer, 2018.

[7] Pauline L Baniqued, Courtney L Gallen, Michelle W Voss, Agnieszka Z Burzynska, Chelsea N Wong, Gillian E Cooke, Kristin Duffy, Jason Fanning, Diane K Ehlers, Elizabeth A Salerno, et al. Brain network modularity predicts exercise-related executive function gains in older adults. *Frontiers in aging neuroscience*, 9:426, 2018.

[8] Runxue Bao, Bin Gu, and Heng Huang. Fast oscar and owl regression via safe screening rules. In *International Conference on Machine Learning*, pages 653–663. PMLR, 2020.

[9] Joe Bathelt, Helen O'Reilly, Jonathan D Clayden, J Helen Cross, and Michelle de Haan. Functional brain network organisation of children between 2 and 5 years derived from reconstructed activity of cortical sources of high-density eeg recordings. *NeuroImage*, 82:595–604, 2013.

[10] Roger E Beaty, Yoed N Kenett, Alexander P Christensen, Monica D Rosenberg, Mathias Benedek, Qunlin Chen, Andreas Fink, Jiang Qiu, Thomas R Kwapil, Michael J Kane, et al. Robust prediction of individual creative ability from brain functional connectivity. *Proceedings of the National Academy of Sciences*, 115(5):1087–1092, 2018.

[11] Timothy EJ Behrens, H Johansen Berg, Saad Jbabdi, Matthew FS Rushworth, and Mark W Woolrich. Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? *neuroimage*, 34(1):144–155, 2007.

[12] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.

[13] Esta A Berg. A simple objective technique for measuring flexibility in thinking. *The Journal of general psychology*, 39(1):15–22, 1948.

[14] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[15] Alaa Bessadok, Mohamed Ali Mahjoub, and Islem Rekik. Hierarchical adversarial connectomic domain alignment for target brain graph prediction and classification from a source graph. In *International Workshop on PRedictive Intelligence In MEdicine*, pages 105–114. Springer, 2019.

[16] Alaa Bessadok, Mohamed Ali Mahjoub, and Islem Rekik. Symmetric dual adversarial connectomic domain alignment for predicting isomorphic brain graph from a baseline graph. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 465–474. Springer, 2019.

[17] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.

[18] Ulrik Brandes, Marco Gaertler, and Dorothea Wagner. Experiments on graph clustering algorithms. In *European Symposium on Algorithms*, pages 568–579. Springer, 2003.

[19] Urs Braun, Sarah F Muldoon, and Danielle S Bassett. On human brain networks in health and disease. *eLS*, pages 1–9, 2015.

[20] Steven L Bressler and Vinod Menon. Large-scale brain networks in cognition: emerging methods and principles. *Trends in cognitive sciences*, 14(6):277–290, 2010.

[21] Colin J Brown, Kathleen P Moriarty, Steven P Miller, Brian G Booth, Jill G Zwicker, Ruth E Grunau, Anne R Synnes, Vann Chau, and Ghassan Hamarneh. Prediction of brain network age and factors of delayed maturation in very preterm infants. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 84–91. Springer, 2017.

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

[23] Ed Bullmore and Olaf Sporns. The economy of brain network organization. *Nature reviews neuroscience*, 13(5):336–349, 2012.

[24] Fernando Calamante. Track-weighted imaging methods: extracting information from a streamlines tractogram. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 30(4):317–335, 2017.

[25] Raffaella Calati, Jerome J Maller, Chantal Meslin, Jorge Lopez-Castroman, Karen Ritchie, Philippe Courtet, and Sylvaine Artero. Repatriation is associated with isthmus cingulate cortex reduction in community-dwelling elderly. *The World Journal of Biological Psychiatry*, 19(6):421–430, 2018.

[26] Vince D Calhoun and Jing Sui. Multimodal fusion of brain imaging data: a key to finding the missing link (s) in complex mental illness. *Biological psychiatry: cognitive neuroscience and neuroimaging*, 1(3):230–244, 2016.

[27] Bokai Cao, Lifang He, Xiaokai Wei, Mengqi Xing, Philip S Yu, Heide Klumpp, and Alex D Leow. t-bne: Tensor-based brain network embedding. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 189–197. SIAM, 2017.

[28] Gustavo Carlo, Marcela Raffaelli, Deborah J Laible, and Kathryn A Meyer. Why are girls less physically aggressive than boys? personality and parenting mediators of physical aggression. *Sex Roles*, 40(9):711–729, 1999.

[29] Dorwin Cartwright and Frank Harary. Structural balance: a generalization of heider's theory. *Psychological review*, 63(5):277, 1956.

[30] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark Hasegawa-Johnson, and Thomas S Huang. Dilated recurrent neural networks. *arXiv preprint arXiv:1710.02224*, 2017.

[31] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.

[32] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[33] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[34] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[35] Tao Chen, Zhi Li, Ji-fang Cui, Jia Huang, Muireann Irish, Ya Wang, and Raymond CK Chan. The neural substrates of sex differences in balanced time perspective: A unique role for the precuneus. *Brain Imaging and Behavior*, pages 1–9, 2022.

[36] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 433–442, 2019.

[37] Sean D Conrin, Liang Zhan, Zachery D Morrissey, Mengqi Xing, Angus Forbes, Pauline Maki, Mohammed R Milad, Olusola Ajilore, Scott A Langenecker, and Alex D Leow. From default mode network to the basal configuration: sex differences in the

resting-state brain connectivity as a function of age and their clinical correlates. *Frontiers in Psychiatry*, 9:365, 2018.

[38] Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. Brainnnexplainer: An interpretable graph neural network framework for brain network based disease analysis. *arXiv preprint arXiv:2107.05097*, 2021.

[39] Zhihua Cui, Fei Xue, Xingjuan Cai, Yang Cao, Gai-ge Wang, and Jinjun Chen. Detection of malicious code variants based on deep learning. *IEEE Transactions on Industrial Informatics*, 14(7):3187–3196, 2018.

[40] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning*, pages 2702–2711, 2016.

[41] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In *International conference on machine learning*, pages 1115–1124. PMLR, 2018.

[42] Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 929–934. IEEE, 2018.

[43] Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980, 2006.

[44] Christophe Destrieux, Bruce Fischl, Anders Dale, and Eric Halgren. Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. *Neuroimage*, 53(1):1–15, 2010.

[45] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.

[46] Niharika Shimona Dsouza, Mary Beth Nebel, Deana Crocetti, Joshua Robinson, Stewart Mostofsky, and Archana Venkataraman. M-gcn: A multimodal graph convolutional network to integrate functional and structural connectomics data to predict multidimensional phenotypic characterizations. In *Medical Imaging with Deep Learning*, pages 119–130. PMLR, 2021.

[47] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[48] Martin Dyrba, Michel Grothe, Thomas Kirste, and Stefan J Teipel. Multimodal analysis of functional and structural disconnection in a lzheimer's disease using multiple kernel svm. *Human brain mapping*, 36(6):2118–2131, 2015.

[49] Tom Eichele, Stefan Debener, Vince D Calhoun, Karsten Specht, Andreas K Engel, Kenneth Hugdahl, D Yves Von Cramon, and Markus Ullsperger. Prediction of human errors by maladaptive changes in event-related brain networks. *Proceedings of the National Academy of Sciences*, 105(16):6173–6178, 2008.

[50] Barbara A Eriksen and Charles W Eriksen. Effects of noise letters upon the identification of a target letter in a nonsearch task. *Perception & psychophysics*, 16(1):143–149, 1974.

[51] Deng-Ping Fan, Tao Zhou, Ge-Peng Ji, Yi Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Inf-net: Automatic covid-19 lung infection segmentation from ct images. *IEEE Transactions on Medical Imaging*, 39(8):2626–2637, 2020.

[52] Yicheng Fang, Huangqi Zhang, Jicheng Xie, Minjie Lin, Lingjun Ying, Peipei Pang, and Wenbin Ji. Sensitivity of chest ct for covid-19: comparison to rt-pcr. *Radiology*, 296(2):E115–E117, 2020.

[53] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[54] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[55] Holger Finger, Marlene Bönstrup, Bastian Cheng, Arnaud Messé, Claus Hilgetag, Götz Thomalla, Christian Gerloff, and Peter König. Modeling of large-scale functional brain networks based on structural connectivity from dti: comparison with eeg derived phase coupling networks and evaluation of alternative methods along the modeling path. *PLoS computational biology*, 12(8):e1005025, 2016.

[56] Florian U Fischer, Dominik Wolf, Armin Scheurich, and Andreas Fellgiebel. Association of structural global brain network properties with intelligence in normal aging. *PloS one*, 9(1):e86258, 2014.

[57] Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.

[58] Alex Fornito, Andrew Zalesky, and Edward Bullmore. *Fundamentals of brain network analysis*. Academic Press, 2016.

[59] Igor Fortel, Laura E Korthauer, Zachery Morrissey, Liang Zhan, Olusola Ajilore, Ouri Wolfson, Ira Driscoll, Dan Schonfeld, and Alex Leow. Connectome signatures of hyperexcitation in cognitively intact middle-aged female apoe-ε4 carriers. *Cerebral Cortex*, 30(12):6350–6362, 2020.

[60] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[61] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019.

[62] Hongyang Gao and Shuiwang Ji. Graph u-nets. *arXiv preprint arXiv:1905.05178*, 2019.

[63] Bao Ge, Lei Guo, Tuo Zhang, Xintao Hu, Junwei Han, and Tianming Liu. Resting state fmri-guided fiber clustering: methods and applications. *Neuroinformatics*, 11(1):119–133, 2013.

[64] Sarah Genon, Andrew Reid, Robert Langner, Katrin Amunts, and Simon B Eickhoff. How to characterize the function of a brain region. *Trends in cognitive sciences*, 22(4):350–364, 2018.

[65] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[66] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[67] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[68] Kaundinya Gopinath, Venkatagiri Krishnamurthy, Romeo Cabanban, and Bruce A Crosson. Hubs of anticorrelation in high-resolution resting-state functional connectivity network architecture. *Brain Connectivity*, 5(5):267–275, 2015.

[69] Hong Gu, Yuzheng Hu, Xi Chen, Yong He, and Yihong Yang. Regional excitation-inhibition balance predicts default-mode network deactivation via functional connectivity. *Neuroimage*, 185:388–397, 2019.

[70] Anne Hafkemeijer, Christiane Möller, Elise GP Dopper, Lize C Jiskoot, Tijn M Schouten, John C Van Swieten, Wiesje M Van der Flier, Hugo Vrenken, Yolande AL Pijnenburg, Frederik Barkhof, et al. Resting state functional connectivity differences between behavioral variant frontotemporal dementia and alzheimer's disease. *Frontiers in human neuroscience*, 9:474, 2015.

[71] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[72] Xuejun Hao, Dongrong Xu, Ravi Bansal, Zhengchao Dong, Jun Liu, Zhishun Wang, Alayar Kangarlu, Feng Liu, Yunsuo Duan, Satie Shova, et al. Multimodal magnetic resonance imaging: The coordinated use of multiple, mutually informative probes to understand brain structure and function. *Human brain mapping*, 34(2):253–271, 2013.

[73] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126. PMLR, 2020.

[74] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[75] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[76] Fritz Heider. Attitudes and cognitive organization. *The Journal of psychology*, 21(1):107–112, 1946.

[77] Claus C Hilgetag and Alexandros Goulas. 'hierarchy'in the organization of brain networks. *Philosophical Transactions of the Royal Society B*, 375(1796):20190319, 2020.

[78] Lucy V Hiscox, Curtis L Johnson, Matthew DJ McGarry, Helen Marshall, Craig W Ritchie, Edwin JR Van Beek, Neil Roberts, and John M Starr. Mechanical property alterations across the cerebral cortex due to alzheimer's disease. *Brain communications*, 2(1):fcz049, 2020.

[79] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[80] Jonas Hornung, Elke Smith, Jessica Junger, Katharina Pauly, Ute Habel, and Birgit Derntl. Exploring sex differences in the neural correlates of self-and other-referential gender stereotyping. *Frontiers in behavioral neuroscience*, 13:31, 2019.

[81] Yifan Hou, Jian Zhang, James Cheng, Kaili Ma, Richard T. B. Ma, Hongzhi Chen, and Ming-Chang Yang. Measuring and improving the use of graph information in graph neural networks. In *International Conference on Learning Representations*, 2020.

[82] Yifan Hou, Jian Zhang, James Cheng, Kaili Ma, Richard TB Ma, Hongzhi Chen, and Ming-Chang Yang. Measuring and improving the use of graph information in graph neural networks. In *International Conference on Learning Representations*, 2019.

[83] Chenhui Hu, Ronghui Ju, Yusong Shen, Pan Zhou, and Quanzheng Li. Clinical decision support for alzheimer's disease based on deep learning and brain network. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2016.

[84] Hanzhe Hu, Deyi Ji, Weihao Gan, Shuai Bai, Wei Wu, and Junjie Yan. Class-wise dynamic graph convolution for semantic segmentation. In A Vedaldi, H Bischof, T Brox, and Frahm JM., editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 12362. Springer, Cham, November 2020.

[85] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[86] Hai K Huang. *PACS and imaging informatics: basic principles and applications*. John Wiley & Sons, 2011.

[87] Haiqing Huang and Mingzhou Ding. Linking functional connectivity and structural connectivity quantitatively: a comparison of methods. *Brain connectivity*, 6(2):99–108, 2016.

[88] Jiashuang Huang, Luping Zhou, Lei Wang, and Daoqiang Zhang. Integrating functional and structural connectivities via diffusion-convolution-bilinear neural network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 691–699. Springer, 2019.

[89] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, Dawei Yin, and Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. *arXiv preprint arXiv:2001.06216*, 2020.

[90] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. In *Advances in neural information processing systems*, pages 4558–4567, 2018.

[91] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, pages 603–612, 2019.

[92] Kai Hwang, Michael N Hallquist, and Beatriz Luna. The development of hub architecture in the human functional brain network. *Cerebral Cortex*, 23(10):2380–2393, 2013.

[93] Muhammad U Ilyas, M Zubair Shafiq, Alex X Liu, and Hayder Radha. A distributed and privacy preserving algorithm for identifying information hubs in social networks. In *2011 Proceedings IEEE INFOCOM*, pages 561–565. IEEE, 2011.

[94] Hao Jiang, Peng Cao, MingYi Xu, Jinzhu Yang, and Osmar Zaiane. Hi-gcn: A hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction. *Computers in Biology and Medicine*, 127:104096, 2020.

[95] Ma Jun, Ge Cheng, Wang Yixin, An Xingle, Gao Jiantao, Yu Ziqi, Zhang Minqing, Liu Xin, Deng Xueyuan, Cao Shucheng, Wei Hao, Mei Sen, Yang Xiaoyu, Nie Ziwei, Li Chen, Tian Lu, Zhu Yuntao, Zhu Qiongjie, Dong Guoqiang, and He Jian. COVID-19 CT Lung and Infection Segmentation Dataset, April 2020.

[96] Jinhong Jung, Jaemin Yoo, and U Kang. Signed graph diffusion network. *arXiv preprint arXiv:2012.14191*, 2020.

[97] Leonard Kaufmann. Clustering by means of medoids. In *Proc. Statistical Data Analysis Based on the L1 Norm Conference, Neuchatel, 1987*, pages 405–416, 1987.

[98] Jeremy Kawahara, Colin J Brown, Steven P Miller, Brian G Booth, Vann Chau, Ruth E Grunau, Jill G Zwicker, and Ghassan Hamarneh. Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *NeuroImage*, 146:1038–1049, 2017.

[99] Kristian Kersting, Nils M. Kriege, Christopher Morris, Petra Mutzel, and Marion Neumann. Benchmark data sets for graph kernels, 2016.

[100] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[101] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[103] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[104] Xiangnan Kong and Philip S Yu. Brain network analysis: a data mining perspective. *ACM SIGKDD Explorations Newsletter*, 15(2):30–38, 2014.

[105] Laura E Korthauer, Liang Zhan, Olusola Ajilore, A Leow, and Ira Driscoll. Disrupted topology of the resting state structural connectome in middle-aged apoe $\varepsilon 4$ carriers. *Neuroimage*, 178:295–305, 2018.

[106] Nevena Kraljević, H Lina Schaare, Simon B Eickhoff, Peter Kochunov, BT Thomas Yeo, Shahrzad Kharabian Masouleh, and Sofie L Valk. Behavioral, anatomical and heritable convergence of affect and cognition in superior frontal cortex. *NeuroImage*, 243:118561, 2021.

[107] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[108] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, 2012.

[109] Sofia Ira Ktena, Sarah Parisot, Enzo Ferrante, Martin Rajchl, Matthew Lee, Ben Glocker, and Daniel Rueckert. Metric learning with spectral graph convolutions on brain connectivity networks. *NeuroImage*, 169:431–442, 2018.

[110] M Kutova, J Mrzilkova, J Riedlova, and P Zach. Asymmetric changes in limbic cortex and planum temporale in patients with alzheimer disease. *Current Alzheimer Research*, 15(14):1361–1368, 2018.

[111] Pamela J LaMontagne, Tammie LS Benzinger, John C Morris, Sarah Keefe, Russ Hornbeck, Chengjie Xiong, Elizabeth Grant, Jason Hassenstab, Krista Moulder, Andrei G Vlassenko, et al. Oasis-3: longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease. *MedRxiv*, 2019.

[112] Pamela J. LaMontagne, Sarah Keefe, Wallace Lauren, Chengjie Xiong, Elizabeth A. Grant, Krista L. Moulder, John C. Morris, Tammie L.S. Benzinger, and Daniel S. Marcus. Oasis-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer's disease. *Alzheimer's & Dementia*, 14(7, Supplement):P1097, 2018.

[113] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[114] Chin-Feng Lee, Jau-Ji Shen, Kun-Liang Hou, and Fang-Wei Hsu. A high-performance computing method for photographic mosaics upon the hadoop framework. *Journal of Internet Technology*, 20(5):1343–1358, 2019.

[115] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. *arXiv preprint arXiv:1904.08082*, 2019.

[116] Robert W Levenson, Virginia E Sturm, and Claudia M Haase. Emotional and behavioral symptoms in neurodegenerative disease: a model for studying the neural bases of psychopathology. *Annual review of clinical psychology*, 10:581, 2014.

[117] Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8950–8959, 2020.

[118] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H Staib, Pamela Ventola, and James S Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 74:102233, 2021.

[119] Xingjuan Li, Yu Li, and Xue Li. Predicting clinical outcomes of alzheimer's disease from complex brain networks. In *International Conference on Advanced Data Mining and Applications*, pages 519–525. Springer, 2017.

[120] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In *Advances in Neural Information Processing Systems*, pages 9225–9235, 2018.

[121] Yu Li, Yuan Tian, Jiawei Zhang, and Yi Chang. Learning signed network embedding via graph attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4772–4779, 2020.

[122] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[123] Zhifeng Liang, Jean King, and Nanyin Zhang. Anticorrelated resting-state functional connectivity in awake rat brain. *Neuroimage*, 59(2):1190–1199, 2012.

[124] Jiahao Liu, Guixiang Ma, Fei Jiang, Chun-Ta Lu, S Yu Philip, and Ann B Ragin. Community-preserving graph convolutions for structural and functional joint embedding of brain networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1163–1168. IEEE, 2019.

[125] Yuhang Liu, Fandong Zhang, Qianyi Zhang, Siwen Wang, Yizhou Wang, and Yizhou Yu. Cross-view correspondence reasoning based on bipartite graph convolutional net-

work for mammogram mass detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3812–3822, 2020.

[126] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[127] Qingqing Long, Yiming Wang, Lun Du, Guojie Song, Yilun Jin, and Wei Lin. Hierarchical community structure preserving network embedding: A subspace approach. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 409–418, 2019.

[128] Jinglei Lv, Lei Guo, Xintao Hu, Tuo Zhang, Kaiming Li, Degang Zhang, Jianfei Yang, and Tianming Liu. Fiber-centered analysis of brain connectivities using dti and resting state fmri data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 143–150. Springer, 2010.

[129] Jinglei Lv, Lei Guo, Kaiming Li, Xintao Hu, Dajiang Zhu, Junwei Han, and Tianming Liu. Activated fibers: fiber-centered activation detection in task-based fmri. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 574–587. Springer, 2011.

[130] Guixiang Ma, Nesreen K Ahmed, Theodore L Willke, Dipanjan Sengupta, Michael W Cole, Nicholas B Turk-Browne, and Philip S Yu. Deep graph similarity learning for brain data analysis. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2743–2751, 2019.

[131] Guixiang Ma, Nesreen K Ahmed, Theodore L Willke, and Philip S Yu. Deep graph similarity learning: A survey. *Data Mining and Knowledge Discovery*, 35(3):688–725, 2021.

[132] 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 *2017 IEEE International Conference on Data Mining (ICDM)*, pages 967–972. IEEE, 2017.

[133] Yao Ma, Suhang Wang, Charu C Aggarwal, and Jiliang Tang. Graph convolutional networks with eigenpooling. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 723–731, 2019.

[134] Rossana Mastrandrea, Andrea Gabrielli, Fabrizio Piras, Gianfranco Spalletta, Guido Caldarelli, and Tommaso Gili. Organization and hierarchy of the human functional brain network lead to a chain-like core. *Scientific reports*, 7(1):1–13, 2017.

[135] Marcelo G Mattar and Danielle S Bassett. Brain network architecture. *Network science in cognitive psychology*, page 30, 2019.

[136] MedSeg, Håvard Bjørke Jenssen, and Tomas Sakinis. MedSeg Covid Dataset 1. 1 2021.

[137] David Meunier, Renaud Lambiotte, Alex Fornito, Karen Ersche, and Edward T Bullmore. Hierarchical modularity in human brain functional networks. *Frontiers in neuroinformatics*, 3:37, 2009.

[138] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[139] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[140] Christoph Molnar. *Interpretable Machine Learning*. Lulu. com, 2020.

[141] Oury Monchi, Michael Petrides, Valentina Petre, Keith Worsley, and Alain Dagher. Wisconsin card sorting revisited: distinct neural circuits participating in different stages of the task identified by event-related functional magnetic resonance imaging. *Journal of Neuroscience*, 21(19):7733–7741, 2001.

[142] SP Morozov, AE Andreychenko, NA Pavlov, AV Vladzymyrskyy, NV Ledikhova, VA Gombolevskiy, Ivan A Blokhin, PB Gelezhe, AV Gonchar, and V Yu Chernina. Mosmeddata: Chest ct scans with covid-19 related findings dataset. *arXiv preprint arXiv:2005.06465*, 2020.

[143] Christopher Morris, Nils M Kriege, Kristian Kersting, and Petra Mutzel. Faster kernels for graphs with continuous attributes via hashing. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1095–1100. IEEE, 2016.

[144] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-

order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4602–4609, 2019.

[145] Nicholas A Nystrom, Michael J Levine, Ralph Z Roskies, and J Ray Scott. Bridges: a uniquely flexible hpc resource for new communities and data analytics. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, pages 1–8, 2015.

[146] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[147] Francesco Orsini, Paolo Frasconi, and Luc De Raedt. Graph invariant kernels. In *Proceedings of the twenty-fourth international joint conference on artificial intelligence*, volume 2015, pages 3756–3762. IJCAI-INT JOINT CONF ARTIF INTELL, 2015.

[148] Verna C Pangman, Jeff Sloan, and Lorna Guse. An examination of psychometric properties of the mini-mental state examination and the standardized mini-mental state examination: implications for clinical practice. *Applied Nursing Research*, 13(4):209–213, 2000.

[149] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[150] 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.

[151] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10772–10781, 2019.

[152] Ailiang Qi, Dong Zhao, Fanhua Yu, Ali Asghar Heidari, Zongda Wu, Zhennao Cai, Fayadh Alenezi, Romany F Mansour, Huiling Chen, and Mayun Chen. Directional mutation and crossover boosted ant colony optimization with application to covid-19 x-ray image segmentation. *Computers in biology and medicine*, 148:105810, 2022.

[153] Shouliang Qi, Stephan Meesters, Klaas Nicolay, Bart M ter Haar Romeny, and Pauly Ossenblok. The influence of construction methodology on structural brain network measures: A review. *Journal of neuroscience methods*, 253:170–182, 2015.

[154] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 2020.

[155] Yu Qiu, Yun Liu, Shijie Li, and Jing Xu. MiniSeg: An extremely minimum network for efficient COVID-19 segmentation. In *AAAI Conference on Artificial Intelligence*, 2021.

[156] Javier Rasero, Nicola Amoroso, Marianna La Rocca, Sabina Tangaro, Roberto Bellotti, Sebastiano Stramaglia, and Alzheimer's Disease Neuroimaging Initiative. Multivariate regression analysis of structural mri connectivity matrices in alzheimer's disease. *PLoS One*, 12(11):e0187281, 2017.

[157] Kaspar Riesen and Horst Bunke. Iam graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 287–297. Springer, 2008.

[158] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[159] Henry Rusinek, Susan De Santi, Dina Frid, Wai-Hon Tsui, Chaim Y Tarshish, Antonio Convit, and Mony J de Leon. Regional brain atrophy rate predicts future cognitive decline: 6-year longitudinal mr imaging study of normal aging. *Radiology*, 229(3):691–696, 2003.

[160] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[161] Mert R Sabuncu and Ender Konukoglu. Clinical prediction from structural brain mri scans: a large-scale empirical study. *Neuroinformatics*, 13(1):31–46, 2015.

[162] Russell P Sawyer, Federico Rodriguez-Porcel, Matthew Hagen, Rhonna Shatz, and Alberto J Espay. Diagnosing the frontal variant of alzheimer's disease: a clinician's yellow brick road. *Journal of clinical movement disorders*, 4(1):1–9, 2017.

[163] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[164] Sambu Seo, Johannes Mohr, Anne Beck, Torsten Wüstenberg, Andreas Heinz, and Klaus Obermayer. Predicting the future relapse of alcohol-dependent patients from structural and functional brain images. *Addiction biology*, 20(6):1042–1055, 2015.

[165] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[166] 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.

[167] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[168] José M Soares, Paulo Marques, Victor Alves, and Nuno Sousa. A hitchhiker's guide to diffusion tensor imaging. *Frontiers in neuroscience*, 7:31, 2013.

[169] Roger D Soberanis-Mukul, Nassir Navab, and Shadi Albarqouni. Uncertainty-based graph convolutional networks for organ segmentation refinement. In *Medical Imaging with Deep Learning*, pages 755–769. PMLR, 2020.

[170] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

[171] Stamatios N Sotiropoulos and Andrew Zalesky. Building connectomes using diffusion mri: why, how and but. *NMR in Biomedicine*, 32(4):e3752, 2019.

[172] Victor Spirin and Leonid A Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the national Academy of sciences*, 100(21):12123–12128, 2003.

[173] Olaf Sporns. The human connectome: origins and challenges. *Neuroimage*, 80:53–61, 2013.

[174] Abubakhari Sserwadda and Islem Rekik. Topology-guided cyclic brain connectivity generation using geometric deep learning. *Journal of Neuroscience Methods*, 353:108988, 2021.

[175] Catherine J Stoodley, Eve M Valera, and Jeremy D Schmahmann. Functional topography of the cerebellum for motor and cognitive tasks: an fmri study. *Neuroimage*, 59(2):1560–1570, 2012.

[176] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

[177] Hang Su, Dong Zhao, Hela Elmannai, Ali Asghar Heidari, Sami Bourouis, Zongda Wu, Zhennao Cai, Wenyong Gui, and Mayun Chen. Multilevel threshold image segmentation for covid-19 chest radiography: A framework using horizontal and vertical multiverse optimization. *Computers in Biology and Medicine*, 146:105618, 2022.

[178] Hang Su, Dong Zhao, Fanhua Yu, Ali Asghar Heidari, Yu Zhang, Huiling Chen, Chengye Li, Jingye Pan, and Shichao Quan. Horizontal and vertical search artificial bee colony for image segmentation of covid-19 x-ray images. *Computers in Biology and Medicine*, 142:105181, 2022.

[179] Jing Sui, Godfrey Pearlson, Arvind Caprihan, Tülay Adali, Kent A Kiehl, Jingyu Liu, Jeremy Yamamoto, and Vince D Calhoun. Discriminating schizophrenia and bipolar disorder by fusing fmri and dti in a multimodal cca+ joint ica model. *Neuroimage*, 57(3):839–855, 2011.

[180] Hui Sun, Wenjing Zhang, Hengyi Cao, Huaiqiang Sun, Jing Dai, Siyi Li, Jiaxin Zeng, Xia Wei, Biqiu Tang, Qiyong Gong, et al. Linked brain connectivity patterns with psychopathological and cognitive phenotypes in drug-naïve first-episode schizophrenia. *Psychoradiology*, 2(2):43–51, 2022.

[181] Jeffrey J Sutherland, Lee A O'brien, and Donald F Weaver. Spline-fitting with a genetic algorithm: A method for developing classification structure- activity relationships. *Journal of chemical information and computer sciences*, 43(6):1906–1915, 2003.

[182] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[183] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. 2013.

[184] Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:1902.10461*, 2019.

[185] Haoteng Tang, Lei Guo, Xiyao Fu, Benjamin Qu, Paul M Thompson, Heng Huang, and Liang Zhan. Hierarchical brain embedding using explainable graph learning. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2022.

[186] Haoteng Tang, Guixiang Ma, Yurong Chen, Lei Guo, Wei Wang, Bo Zeng, and Liang Zhan. Adversarial attack on hierarchical graph pooling neural networks. *arXiv preprint arXiv:2005.11560*, 2020.

[187] Haoteng Tang, Guixiang Ma, Lei Guo, Xiyao Fu, Heng Huang, and Liang Zhang. Contrastive brain network learning via hierarchical signed graph pooling model. *arXiv preprint arXiv:2207.07650*, 2022.

[188] Haoteng Tang, Guixiang Ma, Lifang He, Heng Huang, and Liang Zhan. Commpool: An interpretable graph pooling framework for hierarchical graph representation learning. *Neural Networks*, 143:669–677, 2021.

[189] Tom N Tombaugh and Nancy J McIntyre. The mini-mental state examination: a comprehensive review. *Journal of the American Geriatrics Society*, 40(9):922–935, 1992.

[190] John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D Peterson, et al. Xsede: Accelerating scientific discovery computing in science & engineering, 16 (5): 62–74, sep 2014. *URL https://doi. org/10.1109/mcse*, 2014.

[191] Nathalie Tzourio-Mazoyer, Brigitte Landeau, Dimitri Papathanassiou, Fabrice Crivello, Octave 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.

[192] Kâmil Uludağ and Alard Roebroeck. General overview on the merits of multimodal neuroimaging data fusion. *Neuroimage*, 102:3–10, 2014.

[193] Martijn P Van Den Heuvel, René S Kahn, Joaquín Goñi, and Olaf Sporns. High-cost, high-capacity backbone for global brain communication. *Proceedings of the National Academy of Sciences*, 109(28):11372–11377, 2012.

[194] Martijn P van den Heuvel and Olaf Sporns. Network hubs in the human brain. *Trends in cognitive sciences*, 17(12):683–696, 2013.

[195] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.

[196] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

[197] Frank Van Overwalle, Qianying Ma, and Elien Heleven. The posterior crus ii cerebellum is specialized for social mentalizing and emotional self-experiences: a meta-analysis. *Social Cognitive and Affective Neuroscience*, 15(9):905–928, 2020.

[198] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

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

[200] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[201] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

[202] Chen Wang, Peter W Horby, Frederick G Hayden, and George F Gao. A novel coronavirus outbreak of global health concern. *The Lancet*, 395(10223):470–473, 2020.

[203] Gai-Ge Wang, Mei Lu, Yong-Quan Dong, and Xiang-Jun Zhao. Self-adaptive extreme learning machine. *Neural Computing and Applications*, 27(2):291–303, 2016.

[204] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018.

[205] Yong Wang, Xiaobin Qiao, and Gai-Ge Wang. Architecture evolution of convolutional neural network using monarch butterfly optimization. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–15, 2022.

[206] David E Warren, Natalie L Denburg, Jonathan D Power, Joel Bruss, Eric J Waldron, Haoxin Sun, Steve E Petersen, and Daniel Tranel. Brain network theory can predict whether neuropsychological outcomes will differ from clinical expectations. *Archives of Clinical Neuropsychology*, 32(1):40–52, 2017.

[207] Susan Whitfield-Gabrieli and Alfonso Nieto-Castanon. Conn: a functional connectivity toolbox for correlated and anticorrelated brain networks. *Brain connectivity*, 2(3):125–141, 2012.

[208] Lara M Wierenga, Martijn P van den Heuvel, Sarai Van Dijk, Yvonne Rijks, Marcel A de Reus, and Sarah Durston. The development of brain network architecture. *Human Brain Mapping*, 37(2):717–729, 2016.

[209] Yawen Wu, Zhepeng Wang, Dewen Zeng, Meng Li, Yiyu Shi, and Jingtong Hu. Decentralized unsupervised learning of visual representations. pages 2326–2333, 2022.

[210] Yawen Wu, Dewen Zeng, Zhepeng Wang, Yiyu Shi, and Jingtong Hu. Federated contrastive learning for volumetric medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 367–377. Springer, 2021.

[211] Yawen Wu, Dewen Zeng, Zhepeng Wang, Yiyu Shi, and Jingtong Hu. Distributed contrastive learning for medical image segmentation. *Medical Image Analysis*, 81:102564, 2022.

[212] Yu-Huan Wu, Shang-Hua Gao, Jie Mei, Jun Xu, Deng-Ping Fan, Rong-Guo Zhang, and Ming-Ming Cheng. Jcs: An explainable covid-19 diagnosis system by joint clas-

sification and segmentation. *IEEE Transactions on Image Processing*, 30:3113–3126, 2021.

[213] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[214] 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.

[215] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

[216] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. Infogcl: Information-aware graph contrastive learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[217] Xiaowei Xu, Xiangao Jiang, Chunlian Ma, Peng Du, Xukun Li, Shuangzhi Lv, Liang Yu, Qin Ni, Yanfei Chen, Junwei Su, et al. A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering*, 6(10):1122–1129, 2020.

[218] Jiadong Yan, Yuzhong Chen, Zhenxiang Xiao, Shu Zhang, Mingxin Jiang, Tianqi Wang, Tuo Zhang, Jinglei Lv, Benjamin Becker, Rong Zhang, et al. Modeling spatio-temporal patterns of holistic functional brain networks via multi-head guided attention graph neural networks (multi-head gagnns). *Medical Image Analysis*, 80:102518, 2022.

[219] Yujun Yan, Jiong Zhu, Marlena Duda, Eric Solarz, Chandra Sripada, and Danai Koutra. Groupinn: Grouping-based interpretable neural network for classification of limited, noisy brain data. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 772–782, 2019.

[220] Chun-Hung Yeh, Derek K Jones, Xiaoyun Liang, Maxime Descoteaux, and Alan Connelly. Mapping structural connectivity using diffusion mri: Challenges and opportunities. *Journal of Magnetic Resonance Imaging*, 53(6):1666–1682, 2021.

[221] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in neural information processing systems*, pages 9244–9255, 2019.

[222] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pages 4800–4810, 2018.

[223] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020.

[224] Hao Yuan and Shuiwang Ji. Structpool: Structured graph pooling via conditional random fields. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[225] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. Xgnn: Towards model-level explanations of graph neural networks. *arXiv preprint arXiv:2006.02587*, 2020.

[226] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[227] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[228] Liang Zhan, Lisanne M Jenkins, Ouri E Wolfson, Johnson Jonaris GadElkarim, Kevin Nocito, Paul M Thompson, Olusola A Ajilore, Moo K Chung, and Alex D Leow. The significance of negative correlations in brain connectivity. *Journal of Comparative Neurology*, 525(15):3251–3265, 2017.

[229] Liang Zhan, Yashu Liu, Jiayu Zhou, Jieping Ye, and Paul M Thompson. Boosting classification accuracy of diffusion mri derived brain networks for the subtypes of mild cognitive impairment using higher order singular value decomposition. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 131–135. IEEE, 2015.

[230] Liang Zhan, Jiayu Zhou, Yalin Wang, Yan Jin, Neda Jahanshad, Gautam Prasad, Talia M Nir, Cassandra D Leonardo, Jieping Ye, Paul M Thompson, et al. Comparison of nine tractography algorithms for detecting abnormal structural brain networks in alzheimer's disease. *Frontiers in aging neuroscience*, 7:48, 2015.

[231] Lu Zhang, Li Wang, and Dajiang Zhu. Jointly analyzing alzheimer's disease related structure-function using deep cross-model attention network. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 563–567. IEEE, 2020.

[232] Lu Zhang, Li Wang, and Dajiang Zhu. Recovering brain structural connectivity from functional connectivity via multi-gcn based generative adversarial network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 53–61. Springer, 2020.

[233] Lu Zhang, Li Wang, Dajiang Zhu, Alzheimer's Disease Neuroimaging Initiative, et al. Predicting brain structural network using functional connectivity. *Medical Image Analysis*, 79:102463, 2022.

[234] Lu Zhang, Akib Zaman, Li Wang, Jingwen Yan, and Dajiang Zhu. A cascaded multimodality analysis in mild cognitive impairment. In *International Workshop on Machine Learning in Medical Imaging*, pages 557–565. Springer, 2019.

[235] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[236] Qian Zhang, Zhiyan Wang, Ali Asghar Heidari, Wenyong Gui, Qike Shao, Huiling Chen, Atef Zaguia, Hamza Turabieh, and Mayun Chen. Gaussian barebone salp swarm algorithm with stochastic fractal search for medical image segmentation: a covid-19 case study. *Computers in Biology and Medicine*, 139:104941, 2021.

[237] Wen Zhang, Liang Zhan, Paul Thompson, and Yalin Wang. Deep representation learning for multimodal brain networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 613–624. Springer, 2020.

[238] Yanfu Zhang and Heng Huang. New graph-blind convolutional network for brain connectome data analysis. In *International Conference on Information Processing in Medical Imaging*, pages 669–681. Springer, 2019.

[239] Yanfu Zhang, Liang Zhan, Weidong Cai, Paul Thompson, and Heng Huang. Integrating heterogeneous brain networks for predicting brain disease conditions. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 214–222. Springer, 2019.

[240] Yanfu Zhang, Liang Zhan, Shandong Wu, Paul Thompson, and Heng Huang. Disentangled and proportional representation learning for multi-view brain connectomes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 508–518. Springer, 2021.

[241] Zhen Zhang, Jiajun Bu, Martin Ester, Jianfeng Zhang, Chengwei Yao, Zhi Yu, and Can Wang. Hierarchical graph pooling with structure learning. *arXiv preprint arXiv:1911.05954*, 2019.

[242] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[243] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, pages 267–283, 2018.

[244] Songwei Zhao, Pengjun Wang, Ali Asghar Heidari, Xuehua Zhao, and Huiling Chen. Boosted crow search algorithm for handling multi-threshold image problems with application to x-ray images of covid-19. *Expert Systems with Applications*, 213:119095, 2022.

[245] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. Data augmentation for graph neural networks. *arXiv preprint arXiv:2006.06830*, 2020.

[246] Chuansheng Zheng, Xianbo Deng, Qing Fu, Qiang Zhou, Jiapei Feng, Hui Ma, Wenyu Liu, and Xinggang Wang. Deep learning-based detection for covid-19 from chest ct using weak label. *MedRxiv*, 2020.

[247] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11. Springer, 2018.

[248] Qi Zhu, Jing Yang, Bingliang Xu, Zhenghua Hou, Liang Sun, and Daoqiang Zhang. Multimodal brain network jointly construction and fusion for diagnosis of epilepsy. *Frontiers in Neuroscience*, page 1121, 2021.

[249] 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.

[250] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2847–2856, 2018.

[251] Daniel Zügner and Stephan Günnemann. Certifiable robustness of graph convolutional networks under structure perturbations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1656–1665, 2020.