Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Wei Dai April 12, 2022

Interpretable Brain Network Analysis with Graph Neural Networks

By

Wei Dai

Carl Yang
Advisor

Computer Science and Mathematics

Carl Yang
Advisor

Michelangelo Grigni
Committee Member

Davide Fossati
Committee Member

2022

Interpretable Brain Network Analysis with Graph Neural Networks

By

Wei Dai

Carl Yang

Advisor

An abstract of
A thesis submitted to the Faculty of the
Emory College of Arts and Sciences of Emory University
in partial fulfillment of the requirements for the degree of
Bachelor of Science with Honors

Computer Science and Mathematics

2022

Abstract

Interpretable Brain Network Analysis with Graph Neural Networks
By Wei Dai

Human brains are at the center of complicated neurobiological systems in which neurons, circuits, and systems communicate in mysterious ways. Understanding the brain's structure and functional processes has long been a fascinating topic of study in neuroscience and clinical disease treatment. One of the most often used paradigms in neuroscience is network mapping of the human brain's connections. Graph Neural Networks (GNNs) have lately gained popularity as a tool for representing complicated network data. However, as a deep model, graph neural networks have limited interpretability. In healthcare, decisions are often critical, and it is hard for researchers to trust the model if the model is not explainable. To allow effective use of deep models in healthcare, we present an interpretable model IBGNN for analyzing disorder-specific salient areas of interest and significant linkages.

Another obstacle to the wide use of GNNs in brain network analysis is the difficulty of performance tuning and comparisons. There has not been a systematic study of how different designs of brain networks will affect the performance of GNNs for brain networks. To tune the interpretable model we made, we present BrainGB, a benchmark for brain network analysis with GNNs. We modularize the implementation designs so that different variants of GNNs can be tested. We use the designed framework to conduct extensive experiments and summarize the best practices in GNN designs for brain networks. To support the development of brain network analysis, we host a website at `https://brainnet.us/` with models, tutorials and examples. We maintain an open-source framework for GNN testing and design on brain networks, which is also available on the website. We anticipate that this research will offer valuable empirical evidence as well as insights for future research in this exciting new field.

GNNs are known to have defects like over-smoothing and over-squashing. To further improve the performance of the interpretable model, we further present a transformer based deep model, specifically designed for brain network analysis. To utilize the clustered nature of the brain network, we add a differential pooling layer, which provides enhanced performance and potential interpretability.

Interpretable Brain Network Analysis with Graph Neural Networks

By

Wei Dai

Carl Yang

Advisor

A thesis submitted to the Faculty of the
Emory College of Arts and Sciences of Emory University
in partial fulfillment of the requirements for the degree of
Bachelor of Science with Honors

Computer Science and Mathematics

2022

Acknowledgments

First, I would like to express my sincere gratitude to my advisor, Dr. Carl Yang, for the invaluable guidance throughout this work. In my sophomore year, I joined Emory Graph Mining Group led by Dr. Yang. Dr. Yang is a great advisor who not only guides my research but also helps me with various aspects of the research, including the methodologies and reading strategies. It is so fortunate for me to have Dr. Yang as my advisor.

Second, I would like to thank my mentor Hejie Cui. She provided me with lots of help on all projects and helped me keep up with tight deadlines. Works in collaboration and under the guidance of Hejie are all fascinating.

I would also like to thank my committee members, Dr. Grigni and Dr. Fossati, for providing excellent suggestions for my thesis. The theoretical establishment of the idea is difficult, but thanks to the suggestions by Dr. Grigni, the work is now much more precise and fluent.

Also, I'd like to thank Chris Gu for providing me with valuable support throughout the thesis process. He helped me a lot in getting through this difficult time. The precious emotional support he provided allowed me to move forward and become a better person.

Finally, I would like to thank my parents, who always supported whatever decision I made and supported me both emotionally and financially. They

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A graph is a popular form of structured data as it captures multiple objects and their relationships simultaneously. They are widely used for representing complex systems of related entities [28]. For example, using the Amazon product co-purchasing network [2], one can model the relationships between any of the two products through nearly three million labels. Brain networks are a special kind of graph. In brain networks, the anatomical areas are represented as nodes, while connectivities between regions are represented as links. Recent neuroscience and brain imaging research have come to the conclusion that interconnections between brain areas are essential variables in neural development and disease analysis [16]. As a result, previous works have widely studied brain networks' prediction power for certain diseases and other special traits.

Brain networks feature a smaller number of nodes than most other graphs. Study shows human brains under fMRI images can be divided into multiple functional regions, called Region of Interests (ROI) [13]. Depending on the division criteria, the number of areas differs in size. However, most division methods, like Automated Anatomical Labeling (AAL) [60], and Freesurfer-generated cortical/subcortical gray matter regions [6], divide the human brain into less than 100 regions, a number much smaller than most other graph datasets. This feature makes it easy to make predic-

tions with a small memory footprint while also presenting challenges of extracting useful information from a small sample data.

Brain networks have several traits worth designers of graph neural networks to give special adaptations. For example, in a particular dataset, the node count for a sample is usually fixed. Plus, each node corresponds to a specific Region of Interest, which gives it special meanings. Furthermore, the connections between nodes are weighted, which offers another dimensionality of data that most graph datasets don't have. These traits, if properly used, can be of great help in model design.

Shallow models, such as graph kernels [30] and tensor factorization [41], have been widely researched in previous work on brain network analysis. Deep learning models, on the other hand, have exploded in popularity in the field of machine learning, with promising results in tasks such as image, video, and audio processing. While traditional machine learning models, like recurrent neural networks and convolutional networks (CNNs) [36] can handle grid-like or sequence data, they cannot handle brain networks, for graphs cannot be directly represented by grid or sequence-like structures.

Recently, Graph Neural Networks (GNNs) are getting more and more attention due to their predicting power and their ability to utilize graph data [69]. Graph Convolutional networks first emerged as a promising way to utilize graphical data and extract useful information [34]. However, traditional graph neural networks lack transparency in predictions. This is fatal when it comes to decision-critical areas like disease analysis. While several explanation methods for GNNs have been proposed, the majority of them produce one explanation for each sample. This explanation is not desirable for brain network analysis, as researchers usually want to know the contribution of each region of the brain to the prediction, an important piece of information not provided by the existing explanation networks. It is recognized that subjects having the same disease share similar brain network patterns. Moreover, none of the existing GNN interpretation models utilizes the unique property of brain

networks.

Another defect of traditional GNN models like graph convolutional networks is that they are known to suffer from over-smoothing [25]. Throughout the training process, the model repeatedly aggregates local information from layer to layer, which will result in loss of local information if too many layers are stacked [80]. Besides, it also suffers from over-squashing, and information from distant nodes does not propagate well.

Besides graph neural networks, neural networks from other areas also suffer from similar issues. For example, recurrent neural networks (RNN) exhibit the over-squashing issue. To address these problems, transformers, as introduced by Vaswani et al. in 2017, combines a sequence of decoders and encoders and utilize positional encoding to keep track of the position of an input token in the sequence of input data [62]. This design effectively solves the problem of over-squashing and can handle input of any length. Several ideas have been proposed to generalize the transformer model to the graph neural networks, including the Graphorformer [72] and the Spectral Attention Network (SAN) [35]. As there is no inherent positional structure in graphs, the two models utilize positional encoding, as in the original transformer model, in different creative ways. The Graphorformer model uses spatial encoding, treating the shortest path distance (SPD) between the two nodes as their distances. The SAN, on the other hand, utilizes the spectral decomposition of a graph, treating eigenvalues and eigenvectors as positional encodings. However, a limitation of both models is that they are not able to make use of the properties of the brain networks.

Another obstacle to brain network research development is that there is no unified framework for performance testing, and no known work has been released that indicates what design works the best. Models developed for brain networks are highly customizable. Different node features, message passing mechanisms and pooling strategies can be used, and the combination of them are numerous. It is very

time-consuming and energy inefficient to try all possible combinations.

To fix the interpretability problem, we propose an interpretable framework that provide disorder-specific biomarkers for connectome-based brain disorder analysis. The structure of the model is shown in Figure 3.1. It comprises two modules: a backbone prediction model and an explanation generator. The backbone prediction model is a message-passing GNN that gives special adaptation to brain networks. The explanation generator, on the other hand, learns a globally shared mask to highlight disorder-specific biomarkers. The learned shared mask is then applied to the original data, creating filtered data that only contain connections that are important for predictions. The filtered data is then fed into the backbone model to tune it, allowing it to make more accurate predictions.

To mitigate the over-smoothing and over-squashing problem of GNNs, we further present a transformer model adapted to the brain networks. In our model, the positional encoding is calculated using the node index, which effectively utilizes the brain network's property of a fixed number of nodes. Brain networks are known to be divided into communities, with stronger within-community connections than inter-community connections. To utilize this feature, we added hierarchical pooling to the framework, reducing the node into clusters before making final predictions, further enhancing the performance.

As we finalize the model design, the difficulty in performance testing becomes an issue. To fix this problem once and for all, we propose a unified brain network benchmark framework, which enables researchers in the area of brain networks to test their models using different variants in design, helping them to find the best design that fits their model. We run tests to find the best setting for our backbone model and present all test results for model design reference.

1.1 Problem Definition

Our problem is formulated as follows: Suppose we have a weighted brain network $G = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$, where $\mathcal{V} = \{v_i\}_{i=1}^{n}$ is the node (ROI) set of size n, $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ is the edge set, and $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix describing connection strengths between ROIs, we wish to find a prediction y, a scalar classification, through a GNN model $\mathcal{M}$.

Chapter 2

Background

2.1 Brain Networks

Brain networks are a special kind of graph with the anatomical areas represented as nodes and connectivities between regions defined as links [48]. Brain networks have gotten a lot of attention in recent years in neuroimaging research to better understand human brain architecture across diverse groups of people [57, 76]. Numerous discoveries in neuroscience research suggest that neural circuits are intimately linked to brain processes, with abnormalities in these neural circuits being discovered in diseased people [66, 37].

Brain networks are constructed from different sources, such as Diffusion Tensor Imaging (DTI) and functional Magnetic Resonance Imaging (fMRI) [4, 81]. As a result, an effective study of the brain connectivities of various label groups is crucial for understanding the biological structures and functions of the complex neural system [46, 71, 55]. Previous models of brain networks are primarily shallow, such as graph kernels [30] and tensor factorization [26, 42]. These models have proven to be incapable of modeling the complex graph structures of the brain networks [17].

2.2 Generic GNN Models

GNNs (Graph Neural Networks) have fundamentally changed graph modeling and analysis for real-world networked data [34], knowledge graphs [53], protein or gene interaction networks [70], and recommendation systems [67]. We give an introduction to two types of GNNs: spectral graph neural networks and spatial graph neural networks.

2.2.1 Spectral Graph Neural Networks

The original graph neural network was motivated by CNN. One of the successful first attempts was introduced by Bruna et al. in 2013, a year after the convolutional neural network was introduced [3]. In the paper, they presented two construction methods, one based upon a hierarchical clustering of the domain and another based on the spectrum of the graph Laplacian. The latter became the base of many subsequent works on graph neural networks. In their work, they first revisited the idea of graph Laplacian

$$\mathcal{L} = I - D^{-1/2} W D^{-1/2} \tag{2.1}$$

Given this formula, they defined the smoothness functional vector $||\nabla x||^2_W$ at a node i as

$$\|\nabla x\|^2_W = \sum_i \sum_j W_{ij}[x(i) - x(j)]^2 \ , \tag{2.2}$$

This leads to a question: how can we maximize the smoothness vector? It turns out the smoothest vector is always an eigenvector of the laplacian $\mathcal{L}$. If we know the eigenvector matrix V, the transformation can be defined as:

$$x_{k+1,j} = h\left(V\sum_{i=1}^{f_{k-1}} F_{k,i,j}V^T x_{k,i}\right) \quad (j = 1\ldots f_k)\ , \tag{2.3}$$

where $F_{k,i,j}$ is a diagonal matrix and h is an activation function. The construction looks quite simple and straightforward. However, as pointed out by Zhang et al., this construction requires $O(n^3)$ time to calculate the eigenvectors, a costly computation for large graphs [79].

Many variants have been proposed to address this computational cost issue. One of the most famous variants is called GCN, proposed by Thomas N. Kipf and Max Welling at ICLR 2017 [34]. They simplified the calculation by defining the activation as

$$H^{(l+1)} = \sigma\Big(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\Big)\ . \tag{2.4}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of a graph with added self connections, I_N is the identity matrix, l is the layer index, and $\tilde{D}$, $W^{(l)}$ are weight matrices to be trained. Since the eigenvector computation is avoided, the computational cost is greatly reduced.

In recent years, attention mechanisms are gaining increasing in popularity. Since attention mechanisms put unequal focus on each input, it performs better under noisy data and accepts inputs of almost any variable size. Many attempts to apply attention mechanisms to graph neural networks. One popular model is Graph Attention Network (GAT), designed by Velickovic et al. [63]. In their proposed model, the coefficient is calculated as

$$\alpha_{ij} = \frac{\exp\Big(\text{LeakyReLU}\Big(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i\|\mathbf{W}\vec{h}_j]\Big)\Big)}{\sum_{k\in\mathcal{N}_i}\exp\Big(\text{LeakyReLU}\Big(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i\|\mathbf{W}\vec{h}_k]\Big)\Big)} \tag{2.5}$$

In the formula, $\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j, \mathbf{W}\vec{h}_k$ are weight matrices assigned to each node, in-

dicating "attention," or importance, of each node. This *self-attention* mechanism, similar to Recurrent Neural Networks, allows the model to focus on important nodes. An advantage of this attention model over other similar works is its computational complexity. Since the equation above does not involve eigendecomposition or other costly matrix operations, the computation cost of a feature is linear with respect to the number of nodes and edges.

2.2.2 Spatial Graph Neural Networks

Since the spectral models are dependent on the Laplacian matrix, there are certain limitations. For example, if the eigenfunctions of the two graphs are different, it is hard to generalize the model from one graph to another [79]. As a result, spatial models that are independent of the eigenfunctions are proposed.

Spatial graph neural network models also originated from the classic convolutional neural networks (CNN) [36]. CNN models primarily deal with grid-like data, such as images [79]. These models are usually unsuitable for graphical data, as the neighborhood nodes and spatial order usually differ from sample to sample. In order to solve this issue, Gao et al. proposed a model called Learnable Graph Convolutional Layer (LGCL) [21]. In their paper, the propagation rule is formulated as

$$X_l = g(X_l, A, k)$$

where the A is the adjacency matrix, g(·) performs the k-largest node selection to transform generic graphs to data of grid-like structures. After X_l is formulated as a grid structure, the model then performs a regular 1-D CNN, and c(·) denotes a regular 1-D CNN that aggregates neighboring information and outputs a new feature vector for each node:

$$x_{l+1} = c(X_l)$$

Through the k-largest node selection, the data is generalized into a matrix of fixed size, making it easier to generalize.

2.3 Brain Specific GNN Models

As we discussed above, the brain networks have many features that could be utilized by neural networks. For example, with prior clinical knowledge, we know there may be some latent features in the brain networks that are difficult to be captured by standard neural networks. To capture these latent features, Suk et al. proposed a latent feature representation with a stacked auto-encoder (SAE) [58]. However, the work provides a limited improvement on the classification model itself. One of the first specialized models on the brain networks is *BrainNetCNN* [31] proposed by Kawahara et al. in 2017.

2.3.1 BrainNetCNN

The most important improvement in BrainNetCNN is the three kinds of layers they proposed. They introduced edge-to-edge, edge-to-node and node-to-graph layers, claiming it will better leverage the topological locality of structural brain networks than other models. Graphical data go through all three kinds of layers sequentially, eventually feeding into a fully connected layer for classification or regression tasks. In the *edge-to-edge* (E2E) layer, each edge, represented by a position in the adjacency matrix, is learned and expanded. The output is defined as a filtered adjacency matrix

$$A_{i,j}^{l+1,n} = \sum_{m=1}^{M^l} \sum_{k=1}^{|\Omega|} r_k^{l,m,n} A_{i,k}^{l,m} + c_k^{l,m,n} A_{k,j}^{l,m} \tag{2.6}$$

where c and r are learnable weights of the nth filter.

Subsequent to E2E filters are the *edge-to-node* (E2N) layers. In this layer, the adjacency matrices are squashed into nodes representations, defined as follows

$$a_i^{l+1,n} = \sum_{m=1}^{M^l} \sum_{k=1}^{|\Omega|} r_k^{l,m,n} A_{i,k}^{l,m} + c_k^{l,m,n} A_{k,j}^{l,m} \tag{2.7}$$

The right-hand side of this is exactly the same as the one in the E2E layer. The left side, however, is a one-dimensional vector with a size equal to the size of the node instead of a 2D vector with the same size as the adjacency matrix.

The *node-to-graph* layer, as its name suggests, further reduces the dimension from node representation to graph representation

$$a^{l+1,n} = \sum_{m=1}^{M^l} \sum_{k=1}^{|\Omega|} w_i^{l,m,n} a_i^{l,m} \tag{2.8}$$

This reduces the result a from a vector of node size to a single scalar.

2.3.2 BrainGNN

In 2021, Li et. al proposed another interesting brain network-specialized GNN model called *BrainGNN* [38]. They utilized one of the most important prior knowledge of the brain networks: region of interests (ROIs). As mentioned in the introduction section, each node in a brain network is assigned a specific ROI, and each ROI has a specific clinical meaning. Therefore, utilizing such a feature would be great for training and interpretation of the results.

In order to leverage this information, they proposed *Ra-GConv* layer, defined as follows

$$vec(W_i^{(l)}) = f_{MLP}^{(l)}(r_i) = \Theta_2^{(l)} relu(\Theta_1^{(l)} r_i) + b^{(l)} \tag{2.9}$$

where r_i is node i's regional information. Θ_1, Θ_2 are weight parameters in MLP

and $b^{(l)}$ is the bias term.

The *Ra-GConv* layer, combined with dropout layers in between and MLP layers on each end, defines the *BrainGNN* model. Since it utilizes the ROI information, it outperforms the previous model, *BrainNetCNN*, in the two classification task, as stated by Li et al. Another advantage is that it provides interpretability, as each node (ROI) corresponds to a specific region in a brain.

Chapter 3

Our Models

3.1 Interpretable GCN and GAT based Model

3.1.1 The Backbone Prediction Model

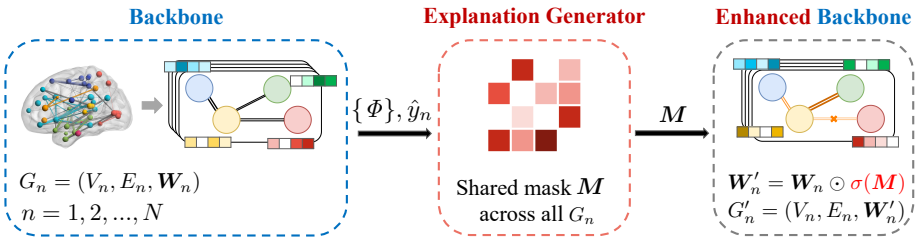


Figure 3.1: An illustration of our message passing GNN model

In a brain network, the edge weights between nodes are formulated by the signal correlation between brain regions. The correlations are not necessarily non-negative, which is a problem for traditional GCNs.

We fixes the problem of negative edge weights and effectively utilizes the edge weights through an special edge-weight-aware message passing algorithm. Specifically, we first construct a message vector $\boldsymbol{m}_{ij} \in \mathbb{R}^d$ for all edges by concatenating

embeddings of a node V_i, its neighbor V_j with its the edge weight w_{ij}:

$$\boldsymbol{m}_{ij}^{(l)} = \text{MLP}\left(\left[\boldsymbol{h}_i^{(l)}; \boldsymbol{h}_j^{(l)}; w_{ij}\right]\right), \tag{3.1}$$

where l is the index of the GNN layer. For each node V_i, we then aggregate messages from all its neighbors $\mathcal{N}_i$ using the following propagation rule:

$$\boldsymbol{h}_i^{(l)} = \xi\left(\sum_{V_j \in \mathcal{N}_i \cup \{V_i\}} \boldsymbol{m}_{ij}^{(l-1)}\right), \tag{3.2}$$

where ξ is a non-linear activation function like ReLU, and $\boldsymbol{h}_i^{(0)}$ is initialized with node feature $\boldsymbol{x}_i$.

After stacking L layers, we employ a readout function summarizing all node embeddings to obtain a graph-level embedding $\boldsymbol{g}$. Formally, we instantiate this function with another MLP and residual connections:

$$\boldsymbol{z} = \sum_{i \in V} \boldsymbol{h}_i^{(L)}, \quad \boldsymbol{g} = \text{MLP}(\boldsymbol{z}) + \boldsymbol{z}. \tag{3.3}$$

We train the model with supervised cross-entropy loss defined as

$$\mathcal{L}_{\text{class}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} -y_{i,j}\log\left(\hat{y}_{i,j}\right),$$

where n is the number of samples and c represents the number of classes.

3.1.2 The Explanation Generator

A popular approach to generate explanations for GNNs is to find an explanation graph G' that maximizes mutual information with the label distribution. The model GNNExplainer proposed by Ying et al. defines the explanation graph G' as a subgraph of G [74]. In some other designs, G' is some alternations of G [44, 77]. Previous

methods either produce a unique explanation for each subject or provide only model-level explanations (e.g., GAT [64]) that cannot offer disease-specific insights. With the unique features of brain networks and the characteristics of disease analysis, a shared explanation graph G' across all samples is more desirable as it captures common patterns for disease-specific analysis.

To address this issue, we design a learnable globally shared edge mask $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ and apply it to individual brain networks across all subjects in a dataset.

Formally, we train the shared edge mask $\boldsymbol{M}$ by maximizing the mutual information between the backbone predictions $\hat{y}$ on the original graph $G = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ and $\hat{y}'$ on the masked graph $G' = (\mathcal{V}, \mathcal{E}, \boldsymbol{W}')$, where $\boldsymbol{W}' = \boldsymbol{W} \odot \sigma(\boldsymbol{M})$. $\odot$ denotes element-wise multiplication, and σ denotes the sigmoid function that standardizes the mask to $[0, 1]^{n \times n}$. This objective of mutual information maximization can be formulated as:

$$\mathcal{L}_{\text{mask}} = \min_{\boldsymbol{M}} - \sum_{i=1}^{c} \mathbb{1}[\hat{y} = i] \log P_{\Phi}\left(\hat{y}' = \hat{y} \mid G'\right),$$

where $P_{\Phi}\left(\hat{y}' = \hat{y} \mid G'\right)$ denotes the conditional probability that the backbone model Φ's prediction $\hat{y}'$ on the masked graph G' is consistent with the prediction $\hat{y}$ on the original graph G.

To encourage the discreteness of the edge weight value in the trained mask, we further apply a sparsity loss

$$\mathcal{L}_{\text{sparsity}} = \sum_{i,j} \boldsymbol{M}_{i,j}$$

defined as the sum of mask parameters to obtain a compact explanation, and another element-wise entropy loss

$$\mathcal{L}_{\text{entropy}} = -(\boldsymbol{M} \log(\boldsymbol{M}) + (1 - \boldsymbol{M}) \log(1 - \boldsymbol{M}))$$

to encourage weight value discreteness in the mask.

Our final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{class}} + \alpha\mathcal{L}_{\text{mask}} + \beta\mathcal{L}_{\text{sparsity}} + \gamma\mathcal{L}_{\text{entropy}},$$

where $\mathcal{L}_{\text{class}}$ is the supervised prediction loss from the backbone model. We normalize the four losses through hyper-parameters α, β and γ so that one loss will not dominate the training process.

After the training is complete, our explanation generator will output an edge mask $\boldsymbol{M}$ that highlights prominent brain network connections for disease predictions. We use the edge mask to investigate disease-specific neurological biomarkers and salient ROIs across all graphs on test datasets.

3.1.3 The Overall Framework

Our model is trained in three stages.

1. Our backbone model is trained on the original graph data.
2. The explanation generator learns a globally-shared edge mask overall training graphs, using the learned backbone model and its prediction as input.
3. We apply the learned global mask $\boldsymbol{M}$ on the original training graphs G to generate filtered graphs G', which are then used to tune the backbone model.

Using this three-step technique, we enhance the prediction model and generate a common explanation mask for model interpretation.

3.2 Brain Transformer

In recent years, transformers have gained increasing popularity because of their performance. We present a transformer model specifically adapted for the brain networks in this work.

3.2.1 Model Structure

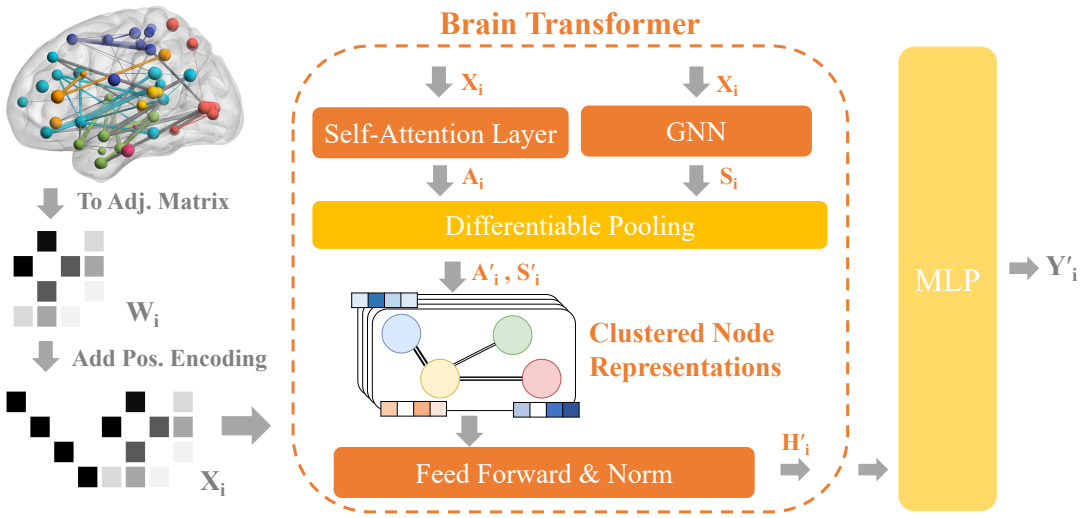


Figure 3.2: An illustration of our transformer model

Our model structure is shown in Figure 3.2. Before feeding into the model, each row of the adjacency matrix W_i is prepended with a node identity vector N_i

$$X_i = [N_i; W_i].$$

whether $X_i \in \mathbb{R}^{n \times 2n}$ is the input of the model. Vector N_i is a one-hot vector of length n with only the value of its node index (row index) being set to one.

X_i is then simultaneously fed into the self-attention layer and the Assignment GNN layer of the Brain Transformer model. The self-attention layer is similar to the "Scaled Dot-Product Attention" detailed in the original transformer model [62]. The

Q_i, K_i, V_i matrices are calculated as

$$Q_i = W_q X_i \tag{3.4}$$

$$K_i = W_k X_i \tag{3.5}$$

$$V_i = W_v X_i \tag{3.6}$$

where $W_q, W_k, W_v \in \mathbb{R}^{2n \times n}$ are learnable weight matrices. Then, the attention is calculated as a normalized product of Q_i, K_i, V_i

$$A_i = Attention(Q, K, V) = softmax(\frac{Q_i K_i^T}{\sqrt{d_{ik}}})V_i$$

where d_{ik} is the dimension of K_i. The Assignment GNN, on the other hand, calculates the assignment matrix $S_i \in \mathbb{R}^{n \times c}$, where c is a hyperparameter that indicates the cluster count we wish the model to find

$$S_i = GNN(x_i)$$

Both the assignment matrix S_i and the attention matrix A_i are fed into the differential pooling layer [73], where S_i and A_i are processed as follows

$$A_i' = softmax(S)^T \cdot A_i \tag{3.7}$$

A layer norm is then applied to the pooled output A_i'

$$H_i' = LayerNorm(A_i')$$

The layer norm is defined as the difference between the input and the expected

value of the input over each dimension, divided by the variance of the input

$$LayerNorm(A_i') = \frac{A_i' - E[A_i']}{\sqrt{Var[A_i'] + \epsilon}} * \gamma + \beta.$$

The γ, β are two learnable parameters, initalized to 1 and 0 respectively [1].

After the layer norm, a 2-layer MLP feed-forward layer follows, with ReLu activation in between

$$H_i' = Linear(ReLu(Linear(H_i)))$$

At last, an MLP layer is followed to generate a graph-level prediction Y_i'

$$Y_i' = MLP(H_i')$$

A performance evaluation of the Brain Transformer model is detailed in Section 5.2.

Chapter 4

Benchmarks and Model Optimizations

As we finalize the model design, performance testing becomes an issue. Graph neural networks feature numerous design variants, and each modification in design can result in a drastic change in the model's performance. To tackle this challenge, we provide a unified brain network benchmark framework, BrainGB, that allows brain network researchers to test their models using a variety of design variants to find the best design for their model. Experiments are conducted to discover the optimal parameters for our backbone model, and the results are presented for model design purposes. The framework is open-sourced, and all instructions are available at `https://brainnet.us/`.

The initialization of ROI features is the first step in applying GNNs to brain networks, followed by the forward pass, which consists of two phases: message passing and pooling. The learned graph-level representation may subsequently be used to analyze brain diseases. We ran experiments on various designs of each part and reported the result in table 4.1.

4.1 Node Feature Construction

The superior performances of graph neural networks are mainly established when there are natural node features for each node in the graph. This is true for many other graphs like social networks, where each node (account) is characterized by its posts, likes and profiles. However, in brain network analysis, natural node properties are not available. Researchers in the graph machine learning area have investigated numerous feasible techniques to initialize node characteristics in order to use GNNs on non-attributed graphs [10, 14]. We tested the following node features and categorized them as positional or structural:

- *Identity*: A unique one-hot feature vector is initialized for each node [75]. We implement this by providing each node with a zero vector of length n (n is the number of nodes in the graph) and set the value at node's index i to 1.

- *Eigen*: An eigen-decomposition is performed on the weighted matrix. The eigenvectors are then used to generate a k-dimensional feature vector for each node [29].

- *Degree*: The degree of each node is obtained as the node feature. This feature captures structural information of brain regions, and the structural similarity of two areas in their immediate vicinity will be partially represented in the initialized node characteristics.

- *Degree profile*: This method utilizes existing local statistical measures on degree profiles [5]. Each feature $\boldsymbol{x}_i$ of node v_i on graph $\mathcal{G}_n$ is computed as

$$\begin{aligned} \boldsymbol{x}_i = [\deg(v_i) \parallel \min(\mathcal{D}_i) \parallel \max(\mathcal{D}_i) \\ \parallel \text{mean}(\mathcal{D}_i) \parallel \text{std}(\mathcal{D}_i)], \end{aligned} \tag{4.1}$$

  where $\mathcal{D}_i = \{\deg(v_i) \mid (i,j) \in \mathcal{E}_n\}$ describes the degree of node v_i's one-hop

neighborhood and $\|$ denotes concatenation.

- *Connection profile*: The node's corresponding row in the adjacency matrix is used. This outputs a n-dimensional vector where n is the number of nodes. For a particular node i, $m_{i,j}$, the feature value at index j is zero if there is no connection between i and j; the value is the edge weight between i and j if there is an connection between the two nodes.

4.2 Message Passing Mechanisms

Message passing GNNs' ability to learn structures lies in their message-passing schemes, in which the node representation is repeatedly updated by collecting neighbor characteristics over local connections. In each step l, the node representation $\boldsymbol{h}_i^l$ is updated through a message vector $\boldsymbol{m}_i^l$ based on

$$\boldsymbol{m}_i^l = \sum_{j \in \mathcal{N}_i} \boldsymbol{m}_{ij} = \sum_{j \in \mathcal{N}_i} M_l \left(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, w_{ij} \right), \tag{4.2}$$

$$\boldsymbol{h}_i^{l+1} = U_l \left(\boldsymbol{h}_i^l, \boldsymbol{m}_i^l \right), \tag{4.3}$$

where $\mathcal{N}_i$ denotes the neighbors of node v_i in graph $\mathcal{G}$, w_{ij} represents the edge weights between node v_i and v_j, M_l is the message function. In addition, U_l here stands for the update function, and the number of running steps L is defined by the number of GNN layers.

Both permutation equivariance and inductive bias may be used to design the message passing mechanism and achieve good generalization on new networks. In the case of brain networks, we primarily focus on message functions that are beneficial for graph-level predictions. We discuss the influence of different message vector $\boldsymbol{m}_{ij}$ designs including:

- *Edge weighted*: The message $\boldsymbol{m}_{ij}$ passed from node v_j to node v_i is the weighted representation of node v_j, that is

$$\boldsymbol{m}_{ij} = \boldsymbol{h}_j \cdot w_{ij} \tag{4.4}$$

  The weight w_{ij} is given by the edge weight between the two nodes. If all edges have equal weights and $w_{ij} = 1/N_i$, this message passing mechanism is equivalent to the original Graph Convolutional Network (GCN) introduced by Kipf and Welling in 2016 [34].

- *Bin concat*: The edge weight of all edges are split into buckets with equal ranges. The index of the bucket the edge assigned to, calling it $\boldsymbol{b}_t$, is then used as an additional edge representation. The bucket representation $\boldsymbol{b}_t$ is then appended to the original node representation $\boldsymbol{h}_j$. A MLP layer is then followed.

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_j \parallel \boldsymbol{b}_t). \tag{4.5}$$

  This message passing mechanism assigns the same representation to edges with similar edge weights.

- *Edge weight concat*: We concatenate the node representation $\boldsymbol{h}_j$ with stacked edge weight $\boldsymbol{w}_{ij}$. We stack them instead of just appending one copy of the edge weight mainly due to the fact that the node representation $\boldsymbol{h}_j$ is usually a vector of a greater size than the edge weight (a scalar). If we use one copy of the edge weight in the message passing, we observe that it does not have enough impact on the overall training process. A MLP layer is then followed

$$\boldsymbol{w}_{ij} = \|_1^d w_{ij} = w_{ij} \parallel w_{ij} \parallel \ \ldots \ \parallel w_{ij}, \tag{4.6}$$

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_j \parallel \boldsymbol{w}_{ij}). \tag{4.7}$$

d is the stacking dimension, which is equal to the dimension of the node representation $\boldsymbol{h}_j$ to ensure they have similar impact on the result.

- *Node edge concat*: We wish to explore whether it is useful to preserve the original representation as a part of the message passing. In this message passing mechanism, we concatenate the source node representation $\boldsymbol{h}_j$ with the target node representation $\boldsymbol{h}_i$, which is then appended by the edge weight w_{ij} between the two nodes. An MLP layer is then followed.

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_i \parallel \boldsymbol{h}_j \parallel w_{ij}). \tag{4.8}$$

  Since the original representation is preserved, this message passing mechanism could potentially mitigate the over-smoothing problem of GNNs.

- *Node concat*: This message passing function is similar to the *node edge concat*, but we removed the edge weight to test the impact of edge weight in message passing:

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_i \parallel \boldsymbol{h}_j). \tag{4.9}$$

4.3 Attention-Enhanced Message Passing

In recent years, the attention mechanism has become increasingly popular [62]. It is based on the fact that while processing enormous volumes of data, human cognitive systems prefer to choose to concentrate on the vital elements as needed while paying little attention to less important parts. Studies from the area of natural language processing [12] and computer vision [24] have shown that attention mechanism can effectively enhance models' efficiency and accuracy. The attention mechanism can also be applied to GNNs. A popular implementation is the Graph Attention Network

introduced by Veličković et al. in 2017 [63].

An important feature of the brain network is the edge weights that represents the correlation between region of interest. However, traditional graph attention mechanisms do not take edge weights into account. In this experiment, we design several attention mechanisms that make use of the edge weights.

In the following equations, the α_{ij}, the attention factor, is calculated from a single-layer feed-forward neural network parameterized by a weight vector $\mathbf{a}$. The result is then followed by the LeakyReLU nonlinearity σ,

$$\alpha_{ij} = \frac{\exp\left(\sigma\left(\boldsymbol{a}^{\top}\left[\boldsymbol{\Theta}\boldsymbol{x}_i \parallel \boldsymbol{\Theta}\boldsymbol{x}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}(i)\cup\{i\}}\exp\left(\sigma\left(\boldsymbol{a}^{\top}\left[\boldsymbol{\Theta}\boldsymbol{x}_i \parallel \boldsymbol{\Theta}\boldsymbol{x}_k\right]\right)\right)}, \tag{4.10}$$

where $\boldsymbol{\Theta}$ represents a learnable linear transformation matrix. This aligns with the attention calculation from the GAT model [63].

- *Attention weighted*: This is the attention mechanism employed by the orginal GAT paper.

$$\boldsymbol{m}_{ij} = \boldsymbol{h}_j \cdot \alpha_{ij} \tag{4.11}$$

- *Edge weighted w/ attn*: In this attention function, the *attention weighted* mechanism from Eq. 4.11 is further weighted by the edge weights between the two nodes.

$$\boldsymbol{m}_{ij} = \boldsymbol{h}_j \cdot \alpha_{ij} \cdot w_{ij} \tag{4.12}$$

- *Attention edge sum*: This is similar to *Edge weighted w/ attn*, except that the

attention weight and the edge weight are added instead of multiplied.

$$\boldsymbol{m}_{ij} = \boldsymbol{h}_j \cdot (\alpha_{ij} + w_{ij}) \tag{4.13}$$

- *Node edge concat w/ attn*: We enhance the *node edge concat* message passing mechanism with attentions. The source node representation is weighted by the attention function. An MLP layer is followed.

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_i \parallel (\boldsymbol{h}_j \cdot \alpha_{ij}) \parallel w_{ij}). \tag{4.14}$$

- *Node concat w/ attn*: Similar to the message passing counterpart, we remove the edge weight in the *Node edge concat w/ attn* function to see the impact of edge weights. We multiply the attention score α_{ij} between node v_i and node v_j with the node representation $\boldsymbol{h}_j$. A MLP layer is then followed.

$$\boldsymbol{m}_j = \text{MLP}(\boldsymbol{h}_i \parallel (\boldsymbol{h}_j \cdot \alpha_{ij})). \tag{4.15}$$

4.4 Pooling Strategies

In the second phase of GNNs, the model computes a feature vector for the whole graph $\boldsymbol{g}_n$ using the pooling strategy R, where

$$\boldsymbol{g}_n = R\left(\{\boldsymbol{h}_i \mid v_i \in \mathcal{G}_n\}\right). \tag{4.16}$$

The pooling function R is similar to the pooling function employed in the convolutional neural networks. It is independent of the message passing mechanism in the previous stage. We compare three popular pooling operators and compare their performances [23, 47]:

- *Mean pooling*: The node features are averaged to obtain the graph embedding. For each single graph $\mathcal{G}_n$, the graph-level representation is computed as

$$\boldsymbol{g}_n = \frac{1}{M} \sum_{k=1}^{M} \boldsymbol{h}_k. \tag{4.17}$$

  where M is the number of nodes in the graph.

- *Sum pooling*: The node features are added to obtain the graph embedding. For each single graph $\mathcal{G}_n$, the graph-level representation is computed as

$$\boldsymbol{g}_n = \sum_{k=1}^{M} \boldsymbol{h}_k. \tag{4.18}$$

- *Concat pooling*: The node features are concatenated to obtain the graph embedding. For each single graph $\mathcal{G}_n$, the graph-level representation is computed as

$$\boldsymbol{g}_n = \|_{k=1}^{M} \boldsymbol{h}_i = \boldsymbol{h}_1 \parallel \boldsymbol{h}_2 \parallel \ \ldots \ \parallel \boldsymbol{h}_k. \tag{4.19}$$

We also utilized the hierarchical pooling [73] in the transformer experiments. However, we are not including it in this experiment due to the great difference between the designs, which makes modular implementation and fair comparison difficult.

4.5 Datasets

We tested all model designs on four datasets: HIV, PNC, ABCD and PPMI. The PPMI dataset is constructed from Diffusion Magnetic Resonance Imaging (dMRI), while the rest three are constructed using the Functional magnetic resonance imaging (fMRI). All datasets are publicly available for download, either directly or on request.

- *Human Immunodeficiency Virus Infection (HIV):* This dataset is collected from the Chicago Early HIV Infection Study at Northwestern University. The dataset includes fMRI imaging of 70 subjects, split equally into 35 early HIV patients and 35 negative controls. The preprocessing includes realignment to the first volume, followed by normalization, slice timing correction, spatial smoothness, band-pass filtering, and linear trend removal of the time series. We focus on the 116 anatomical ROIs [61] and extract a sequence of time courses from them. Finally, the cerebellum part is filtered out, and brain networks with 90 cerebral regions are constructed, with links representing the correlations between ROIs.

- *Philadelphia Neuroimaging Cohort (PNC)*[1]*:* This fMRI dataset is from the Brain Behavior Laboratory at the University of Pennsylvania and the Children's Hospital of Philadelphia. 289 (57.46%) of the 503 included subjects are female, indicating this dataset is balanced across genders. The regions are parcellated based on the 264-node atlas defined by Power et al. [50]. The preprocessing includes slice timing correction, normalization, motion correction, removal of linear trends, bandpass filtering, and spatial smoothing. In the resulting data, each sample contains 264 nodes with time-series data collected through 120 time steps. We focus on the 232 nodes in the Power's atlas associated with major resting-state functional modules [56].

[1] `https://www.nitrc.org/projects/pnc/`

- *Parkinson's Progression Markers Initiative (PPMI)*[2]*:* This dataset is a part of a collaborative study for Parkinson's Research to improve PD therapeutics. We consider the DTI acquisition of 754 subjects, 596 of which are Parkinson's disease patients while 158 are healthy controls. The raw data are first aligned to correct for head motion and eddy current distortions. Then the non-brain parts are removed, and the skull-stripped images are linearly aligned and registered. The brain network is rebuilt using the deterministic 2nd-order Runge-Kutta (RK2) whole-brain tractography technique employing 84 ROIs from T1-weighted structural MRI [78].

- *Adolescent Brain Cognitive Development Study (ABCD)*[3]*:* This research enrolls children aged 9 to 10 years old in 21 locations across the United States. Repeated imaging scans, as well as intensive psychological and cognitive testing, are used to track each child until early adulthood [7]. A total of 7,901 children are involved in the study, with 3,961 (50.1%) of them being female. For the baseline visit, we employ fMRI images, which are processed using the conventional and open-source ABCD-HCP BIDS fMRI Pipeline [4]. After processing, each sample contains a connectivity matrix whose size is 360×360 and BOLD time-series for each node. The region definition is based on the HCP 360 ROI atlas [22].

4.6 Experimental Analysis and Insights

We present experimental data on brain networks constructed from real-world neuroimaging studies using various GNN modular architectures in this part. There are 375 alternative architectures created by varying each design dimension beneath each

[2] `https://www.ppmi-info.org/`
[3] `https://nda.nih.gov/abcd`
[4] `https://github.com/DCAN-Labs/abcd-hcp-pipeline/`

Table 4.1: Performance report (%) of different message passing GNNs

Module	Method	HIV			PNC			PPMI			ABCD		
		Accuracy	F1	AUC	Accuracy	F1	AUC	Accuracy	F1	AUC	Accuracy	F1	AUC
Node Features	*Identity*	50.00$_{\pm 0.00}$	33.33$_{\pm 0.00}$	46.73$_{\pm 10.57}$	57.34$_{\pm 0.17}$	36.44$_{\pm 0.17}$	52.58$_{\pm 4.80}$	79.25$_{\pm 0.24}$	44.21$_{\pm 0.08}$	59.65$_{\pm 6.80}$	49.97$_{\pm 0.13}$	33.32$_{\pm 0.06}$	50.00$_{\pm 0.20}$
	Eigen	65.71$_{\pm 2.86}$	65.45$_{\pm 2.69}$	65.31$_{\pm 2.89}$	51.40$_{\pm 3.92}$	48.63$_{\pm 5.42}$	50.18$_{\pm 7.57}$	74.09$_{\pm 2.77}$	47.36$_{\pm 4.26}$	49.21$_{\pm 1.58}$	50.79$_{\pm 0.82}$	50.79$_{\pm 0.83}$	51.18$_{\pm 1.16}$
	Degree	44.29$_{\pm 5.35}$	35.50$_{\pm 6.10}$	42.04$_{\pm 4.00}$	63.89$_{\pm 2.27}$	59.69$_{\pm 3.85}$	70.25$_{\pm 4.38}$	79.52$_{\pm 2.31}$	49.40$_{\pm 5.17}$	59.73$_{\pm 4.31}$	63.46$_{\pm 1.29}$	63.45$_{\pm 1.28}$	68.16$_{\pm 1.41}$
	Degree profile	50.00$_{\pm 0.00}$	33.33$_{\pm 0.00}$	50.00$_{\pm 0.00}$	51.40$_{\pm 7.21}$	33.80$_{\pm 3.21}$	50.00$_{\pm 0.00}$	77.02$_{\pm 1.97}$	49.45$_{\pm 3.51}$	58.65$_{\pm 2.44}$	49.92$_{\pm 0.11}$	33.30$_{\pm 0.05}$	50.00$_{\pm 0.00}$
	Connection profile	65.71$_{\pm 13.85}$	64.11$_{\pm 13.99}$	**75.10$_{\pm 16.95}$**	69.83$_{\pm 4.15}$	66.20$_{\pm 4.74}$	**76.69$_{\pm 5.04}$**	77.99$_{\pm 2.78}$	52.96$_{\pm 4.52}$	**65.77$_{\pm 4.09}$**	82.42$_{\pm 1.93}$	82.30$_{\pm 2.08}$	**91.33$_{\pm 0.77}$**
Message Passing	*Edge weighted*	50.00$_{\pm 0.00}$	33.33$_{\pm 0.00}$	49.80$_{\pm 4.20}$	64.87$_{\pm 5.44}$	59.70$_{\pm 7.04}$	69.98$_{\pm 4.19}$	79.25$_{\pm 0.24}$	44.21$_{\pm 0.08}$	62.26$_{\pm 2.80}$	74.47$_{\pm 1.17}$	74.36$_{\pm 1.23}$	82.37$_{\pm 1.46}$
	Bin concat	50.00$_{\pm 0.00}$	33.33$_{\pm 0.00}$	49.39$_{\pm 9.25}$	54.74$_{\pm 5.88}$	36.42$_{\pm 3.97}$	61.68$_{\pm 3.91}$	79.25$_{\pm 0.24}$	44.21$_{\pm 0.08}$	52.67$_{\pm 7.16}$	53.72$_{\pm 4.97}$	43.26$_{\pm 12.43}$	61.86$_{\pm 5.79}$
	Edge weight concat	51.43$_{\pm 2.86}$	44.36$_{\pm 6.88}$	48.16$_{\pm 10.13}$	63.68$_{\pm 3.31}$	60.27$_{\pm 5.97}$	67.34$_{\pm 3.02}$	79.25$_{\pm 0.24}$	44.21$_{\pm 0.08}$	59.72$_{\pm 4.65}$	64.59$_{\pm 1.30}$	64.30$_{\pm 1.43}$	70.63$_{\pm 1.02}$
	Node edge concat	65.71$_{\pm 13.85}$	64.11$_{\pm 13.99}$	75.10$_{\pm 16.95}$	69.83$_{\pm 4.15}$	66.20$_{\pm 4.74}$	76.69$_{\pm 5.04}$	77.99$_{\pm 2.78}$	52.96$_{\pm 4.52}$	65.77$_{\pm 4.09}$	82.42$_{\pm 1.93}$	82.30$_{\pm 2.08}$	91.33$_{\pm 0.77}$
	Node concat	70.00$_{\pm 15.91}$	68.83$_{\pm 17.57}$	**77.96$_{\pm 8.20}$**	70.63$_{\pm 2.35}$	67.12$_{\pm 1.81}$	**78.32$_{\pm 1.42}$**	78.41$_{\pm 1.62}$	54.46$_{\pm 3.08}$	**68.34$_{\pm 1.89}$**	80.50$_{\pm 2.27}$	80.10$_{\pm 2.47}$	**91.36$_{\pm 0.92}$**
Message Passing w/ Attention	*Attention weighted*	50.00$_{\pm 0.00}$	33.33$_{\pm 0.00}$	49.80$_{\pm 8.52}$	65.09$_{\pm 2.21}$	60.74$_{\pm 4.89}$	69.79$_{\pm 4.24}$	79.25$_{\pm 0.24}$	44.21$_{\pm 0.08}$	63.24$_{\pm 3.77}$	77.74$_{\pm 0.97}$	77.70$_{\pm 1.01}$	85.10$_{\pm 1.10}$
	Edge weighted w/ attn	50.00$_{\pm 0.00}$	33.33$_{\pm 0.00}$	42.04$_{\pm 15.63}$	62.90$_{\pm 1.22}$	61.14$_{\pm 0.57}$	69.74$_{\pm 2.37}$	79.25$_{\pm 0.24}$	44.21$_{\pm 0.08}$	54.92$_{\pm 4.80}$	78.04$_{\pm 1.96}$	77.81$_{\pm 2.33}$	86.86$_{\pm 0.63}$
	Attention edge sum	51.43$_{\pm 7.00}$	49.13$_{\pm 5.65}$	54.49$_{\pm 15.67}$	61.51$_{\pm 2.86}$	55.36$_{\pm 4.76}$	69.38$_{\pm 3.50}$	79.11$_{\pm 0.40}$	44.17$_{\pm 0.12}$	60.47$_{\pm 6.26}$	75.71$_{\pm 1.52}$	75.59$_{\pm 1.68}$	83.78$_{\pm 0.82}$
	Node edge concat w/ attn	72.86$_{\pm 11.43}$	72.52$_{\pm 11.72}$	78.37$_{\pm 10.85}$	67.66$_{\pm 5.07}$	64.69$_{\pm 5.36}$	74.52$_{\pm 1.20}$	77.30$_{\pm 1.52}$	50.96$_{\pm 4.20}$	63.93$_{\pm 4.89}$	83.10$_{\pm 0.47}$	83.03$_{\pm 0.52}$	**91.85$_{\pm 0.29}$**
	Node concat w/ attn	71.43$_{\pm 9.04}$	70.47$_{\pm 9.26}$	**82.04$_{\pm 11.21}$**	68.85$_{\pm 6.42}$	64.29$_{\pm 10.15}$	**75.36$_{\pm 5.09}$**	78.41$_{\pm 1.43}$	49.98$_{\pm 1.87}$	**68.14$_{\pm 5.01}$**	83.19$_{\pm 0.93}$	83.12$_{\pm 0.96}$	91.55$_{\pm 0.59}$
Pooling Strategies	*Mean pooling*	47.14$_{\pm 15.39}$	41.71$_{\pm 17.36}$	58.78$_{\pm 18.63}$	66.86$_{\pm 2.33}$	61.39$_{\pm 4.88}$	74.20$_{\pm 3.39}$	79.25$_{\pm 0.24}$	44.21$_{\pm 0.08}$	59.64$_{\pm 5.47}$	81.13$_{\pm 0.35}$	81.06$_{\pm 0.34}$	88.49$_{\pm 1.12}$
	Sum pooling	57.14$_{\pm 9.04}$	52.23$_{\pm 12.65}$	57.96$_{\pm 11.15}$	60.13$_{\pm 2.87}$	53.96$_{\pm 7.61}$	66.11$_{\pm 4.22}$	79.39$_{\pm 0.52}$	47.68$_{\pm 3.12}$	61.29$_{\pm 2.11}$	77.48$_{\pm 3.75}$	76.96$_{\pm 4.58}$	87.90$_{\pm 0.65}$
	Concat pooling	65.71$_{\pm 13.85}$	64.11$_{\pm 13.99}$	**75.10$_{\pm 16.95}$**	69.83$_{\pm 4.15}$	66.20$_{\pm 4.74}$	**76.69$_{\pm 5.04}$**	77.99$_{\pm 2.78}$	52.96$_{\pm 4.52}$	**65.77$_{\pm 4.09}$**	82.42$_{\pm 1.93}$	82.30$_{\pm 2.08}$	**91.33$_{\pm 0.77}$**
Other Baselines	BrainNetCNN	60.21$_{\pm 17.16}$	60.12$_{\pm 13.56}$	70.93$_{\pm 4.01}$	71.93$_{\pm 4.90}$	69.94$_{\pm 5.42}$	78.50$_{\pm 3.28}$	77.24$_{\pm 2.09}$	50.24$_{\pm 3.09}$	58.76$_{\pm 8.95}$	85.1$_{\pm 0.92}$	85.7$_{\pm 0.83}$	93.5$_{\pm 0.34}$
	BrainGNN	62.98$_{\pm 11.15}$	60.45$_{\pm 8.96}$	68.03$_{\pm 9.16}$	70.62$_{\pm 4.85}$	68.93$_{\pm 4.01}$	77.53$_{\pm 3.23}$	79.17$_{\pm 1.22}$	44.19$_{\pm 3.11}$	45.26$_{\pm 3.65}$	OOM	OOM	OOM

module. Note that we are not attempting to cover all possible combinations, but rather to rapidly identify a solid design decision for a certain dataset or downstream job. We also compare our modular architecture to two common deep models, BrainNetCNN [32] and BrainGNN [39], for brain networks.

4.6.1 Performance Report

In each subsections below, the other designs of the model are kept as the best setting, concluded from the experiments in other settings.

Node feature

Different node feature initialization methods are experimented with the same model architecture. When test node features, we set node edge concat in Eq. 4.8 as the message passing scheme, and concat pooling in Eq. 4.19 as the pooling strategy.

In our results, the *connection profile* outperforms all other datasets. It surpasses the second best configuration, *degree*, by a margin of 33.99% on the ABCD dataset. The *connection profile* node feature utilizes the edges connected to the node with the edge weights as the inputs. This feature captures the full structural information of the brain network and contains a wealth of information about paired connections that can be utilized to parcellate the brain. We believe this contributes to the sucess of the *connection profile* node feature.

From the result, we conclude that in the field of brain network analysis, structural information of the graph is more important than the positional ones. The the structure node features (e.g., *degree*, *connection profile*) all perform better than the positional ones (e.g. *identity*, *eigen*).

Message passing

Different message passing methods are experimented with the same node feature and pooling method. When testing message passings, we set *connection profile* as the node feature, and *concat pooling* in Eq. 4.19 as the pooling strategy.

Our experiment demonstrates that the *node concat* message passing, detailed in Equation 4.9, performs best across all datasets in terms of AUC. The second best, *node edge concat* (Eq. 4.8), achieves a slightly better result in terms of accuracy and F1 scores on the ABCD dataset. The performance advantage may be the result of the mitigation of the over-smoothing problem of the GNNs. Since the local original representation is preserved, the result of all node embedding being similar becomes unlikely, and the over-smoothing problem is reduced. Out of our expectation, the *node edge concat* performs worse than *node concat*, particularly on smaller datasets, although their performance difference on larger datasets are similar. We suspect this is due to the node feature, *connection profile*, we use. *connection profile* already contains edge weights of all edges connecting to the source node, including the edge weight of the message passing. Therefore, adding another copy of the same edge weight does not bring extra benefit.

Attention-enhanced message passing

Different attention-enhanced message passing methods are tested with the same node feature and pooling methods. We set *connection profile* as the node feature, and *concat pooling* in Eq. 4.19 as the pooling strategy, the same setting as we tested the non-attention message passing functions.

The distribution and order of the five variants are similar to the result in the non-attention message passing. The *node concat w/ attn* (Eq. 4.15) and the *node edge concat w/ attn* (Eq. 4.14) are two best results with similar performance across all four datasets. On average, however, the attention-enhanced version of the message

passing functions performs better than their non-attention counterparts, with up to 5.23% relative improvements at maximum. This confirms the previous findings that the attention mechanisms are effective in GNNs. In addition, in the ABCD dataset, the *node edge concat w/ attn* performs better than the *node concat w/ attn* one, indicating that the additional information of edge weights may be beneficial in large datasets.

Pooling strategies

Different pooling strategies are tested with the same message passing mechanism and pooling method. When we test pooling strategies, we set *connection profile* as the node feature, and *node edge concat* (Eq. 4.8) as the message passing scheme.

Out of all three pooling strategies, we find that the *concat pooling* yields the best result across all datasets. This is likely because that the increased dimensionality of information as a result of concat pooling is beneficial when the model makes a final prediction. With concat pooling, the node representations of all brain nodes are kept and are fed into the final classifier. The other two pooling methods, on the other hands, combines the representation into one node, which result in some loss of information.

Other Baselines

As shown in the result table, BrainGNN requires a greater amount of GPU memory and result in out-of-memory (OOM) on large datasets. Using the best combination, our modular model outperforms both BrainNetCNN and BrainGNN on small datasets (HIV, PPMI) and performs similarly on larger ones. The best combination based on our modular design outperforms both SOTA models of BrainNetCNN and BrainGNN on small datasets (HIV, PPMI) and achieves comparable results with BrainNetCNN on large datsets (PNC, ABCD). These findings highlight the need of carefully ex-

perimenting with our modular GNN designs before moving on to more sophisticated structures that may simply overfit certain datasets.

Chapter 5

Results and Interpretation Analysis

5.1 Experiment Results of Explainable GNN Networks

5.1.1 Datasets and Preprocessings

We evaluate our interpretable framework, namely IBGNN and IBGNN+, using three real-world neuroimaging datasets of different modalities. The two datasets, HIV and PPMI, are already introduced in section 4.5. We introduce the Bipolar dataset in the section below.

- *Bipolar Disorder (BP):* Bipolar Disorder (BP) dataset is collected using the diffusion tensor imaging tractography. The dataset contains 52 bipolar I subjects and 45 healthy controls. The FSL toolbox[1] is used for preprocessing. We employ distortion correction, noise filtering, and repetitive sampling from the distributions of principal diffusion directions for each voxel during the preprocessing

[1] `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/`

Table 5.1: Experimental results (%) of IBGNN on three datasets

Method	HIV			BP			PPMI		
	Accuracy	F1	AUC	Accuracy	F1	AUC	Accuracy	F1	AUC
M2E	$57.14_{\pm 19.17}$	$53.71_{\pm 19.80}$	$57.50_{\pm 18.71}$	$52.56_{\pm 13.86}$	$51.65_{\pm 13.38}$	$52.42_{\pm 13.83}$	$78.69_{\pm 1.78}$	$45.81_{\pm 4.17}$	$50.39_{\pm 2.59}$
MIC	$54.29_{\pm 18.95}$	$53.63_{\pm 19.44}$	$55.42_{\pm 19.10}$	$62.67_{\pm 20.92}$	$63.00_{\pm 21.61}$	$61.79_{\pm 21.74}$	$79.11_{\pm 2.16}$	$49.65_{\pm 5.10}$	$52.39_{\pm 2.94}$
MPCA	$67.14_{\pm 20.25}$	$64.28_{\pm 23.47}$	$69.17_{\pm 20.17}$	$52.56_{\pm 13.12}$	$50.43_{\pm 14.99}$	$52.42_{\pm 13.69}$	$79.15_{\pm 0.57}$	$44.18_{\pm 0.18}$	$50.00_{\pm 0.00}$
MK-SVM	$65.71_{\pm 7.00}$	$62.08_{\pm 7.49}$	$65.83_{\pm 7.41}$	$57.00_{\pm 8.89}$	$41.08_{\pm 13.44}$	$53.75_{\pm 8.00}$	$79.15_{\pm 0.57}$	$44.18_{\pm 0.18}$	$50.00_{\pm 0.00}$
GCN	$70.00_{\pm 12.51}$	$68.35_{\pm 13.28}$	$73.58_{\pm 9.49}$	$55.56_{\pm 13.86}$	$50.71_{\pm 11.75}$	$61.55_{\pm 28.77}$	$78.55_{\pm 1.58}$	$47.87_{\pm 4.40}$	$59.43_{\pm 8.64}$
GAT	$71.43_{\pm 11.66}$	$69.79_{\pm 10.83}$	$77.17_{\pm 9.42}$	$63.34_{\pm 9.15}$	$60.42_{\pm 7.56}$	$67.07_{\pm 5.98}$	$79.02_{\pm 1.25}$	$45.85_{\pm 3.16}$	$64.40_{\pm 6.87}$
PNA	$57.14_{\pm 12.78}$	$45.09_{\pm 19.62}$	$57.14_{\pm 12.78}$	$63.71_{\pm 11.34}$	$55.54_{\pm 14.06}$	$60.30_{\pm 11.89}$	$79.36_{\pm 1.84}$	$51.76_{\pm 10.32}$	$54.71_{\pm 6.77}$
BrainNetCNN	$69.24_{\pm 19.04}$	$67.08_{\pm 11.11}$	$72.09_{\pm 19.01}$	$65.83_{\pm 20.64}$	$64.74_{\pm 17.42}$	$64.32_{\pm 13.72}$	$55.20_{\pm 12.63}$	$\underline{55.45_{\pm 9.15}}$	$52.54_{\pm 10.21}$
BrainGNN	$74.29_{\pm 12.10}$	$73.49_{\pm 10.75}$	$75.00_{\pm 10.56}$	$68.00_{\pm 12.45}$	$62.33_{\pm 13.01}$	$74.20_{\pm 12.93}$	$69.17_{\pm 0.00}$	$44.19_{\pm 0.00}$	$45.26_{\pm 3.65}$
IBGNN	$\underline{82.14_{\pm 10.81}}^{*}$	$\underline{82.02_{\pm 10.86}}^{*}$	$\underline{86.86_{\pm 11.65}}^{*}$	$\underline{73.19_{\pm 12.20}}^{*}$	$\underline{72.87_{\pm 12.09}}^{*}$	$\underline{83.64_{\pm 9.61}}^{*}$	$\mathbf{79.82_{\pm 1.47}}^{*}$	$51.58_{\pm 4.66}$	$\underline{70.65_{\pm 6.55}}^{*}$
IBGNN+	$\mathbf{84.29_{\pm 12.94}}^{*}$	$\mathbf{83.86_{\pm 13.42}}^{*}$	$\mathbf{88.57_{\pm 10.89}}^{*}$	$\mathbf{76.33_{\pm 13.00}}^{*}$	$\mathbf{76.13_{\pm 13.01}}^{*}$	$\mathbf{84.61_{\pm 9.08}}^{*}$	$\underline{79.55_{\pm 1.67}}$	$\mathbf{56.58_{\pm 7.43}}^{*}$	$\mathbf{72.76_{\pm 6.73}}^{*}$

stage. In each sample, 82 region is parcellated based on FreeSurfer-generated cortical/subcortical gray matter regions [51].

5.1.2 Compared Methods

Our interpretable model include two variants: IBGNN and IBGNN+. The IBGNN+ is the variant trained with masked training data, with mask calculated through the explainer. We compare the models with both shallow and deep models. Shallow methods include M2E [41], MPCA [43], MK-SVM [15], and MIC [54]. In MK-SVM, the output graph-level embeddings are evaluated using logistic regression. Deep models include GAT [65], GCN [34] and PNA [9]. We also test state-of-the-art deep models specifically design for brain networks: BrainNetCNN [32] and BrainGNN [39].

5.1.3 Prediction Performance

The results are shown in Table 5.1. Accuracy, F1, and AUC were the metrics we utilized to assess performance. Both of our suggested models outperform the shallow and deep baselines by a significant margin. Compared to shallow models like MIC, our backbone model IBGNN beats them by a wide margin, with BP gains of up to 11% performance advantage. Furthermore, the superiority of our suggested model over previous SOTA deep models supports the usefulness of our brain network-

Table 5.2: Experimental results (%) of Brain Transformer on PPMI Dataset

Method	PPMI		
	Accuracy	F1	AUC
M2E	$78.69_{\pm 1.78}$	$45.81_{\pm 4.17}$	$50.39_{\pm 2.59}$
MIC	$79.11_{\pm 2.16}$	$49.65_{\pm 5.10}$	$52.39_{\pm 2.94}$
MPCA	$79.15_{\pm 0.57}$	$44.18_{\pm 0.18}$	$50.00_{\pm 0.00}$
MK-SVM	$79.15_{\pm 0.57}$	$44.18_{\pm 0.18}$	$50.00_{\pm 0.00}$
GCN	$78.55_{\pm 1.58}$	$47.87_{\pm 4.40}$	$59.43_{\pm 8.64}$
GAT	$79.02_{\pm 1.25}$	$45.85_{\pm 3.16}$	$64.40_{\pm 6.87}$
PNA	$79.36_{\pm 1.84}$	$51.76_{\pm 10.32}$	$54.71_{\pm 6.77}$
BrainNetCNN	$55.20_{\pm 12.63}$	$55.45_{\pm 9.15}$	$52.54_{\pm 10.21}$
BrainGNN	$69.17_{\pm 0.00}$	$44.19_{\pm 0.00}$	$45.26_{\pm 3.65}$
IBGNN	$79.82_{\pm 1.47}$	$51.58_{\pm 4.66}$	$70.65_{\pm 6.55}$
IBGNN+	$79.55_{\pm 1.67}$	$56.58_{\pm 7.43}$	$\mathbf{72.76_{\pm 6.73}}$
BrainTransformer	$\mathbf{81.94_{\pm 1.3}}$	-	$70.12_{\pm 8.40}$

oriented architecture. Moreover, the explanation enhanced model IBGNN+ is able to provide a further performance enhancement of about 3%. IBGNN+ successfully highlights disorder-specific signals while simultaneously benefiting from restricting random noises in particular graphs, as demonstrated by the performance gain achieved by using the explanation generator.

5.2 Experiment Results of Brain Transformer

5.2.1 Performance

The results of the Brain Transformer model are shown in Table 5.2. Accuracy, F1, and AUC were the metrics we utilized to assess performance. The performance of the transformer model is promising. In terms of accuracy, our transformer model outperforms every shallow and deep baseline, with 2.12% advantage over the second place, IBGNN. The AUC, on the other hand, outperforms every baseline except IBGNN+, our enhanced GCN-based explainer framework.

5.3 Neural System Mapping

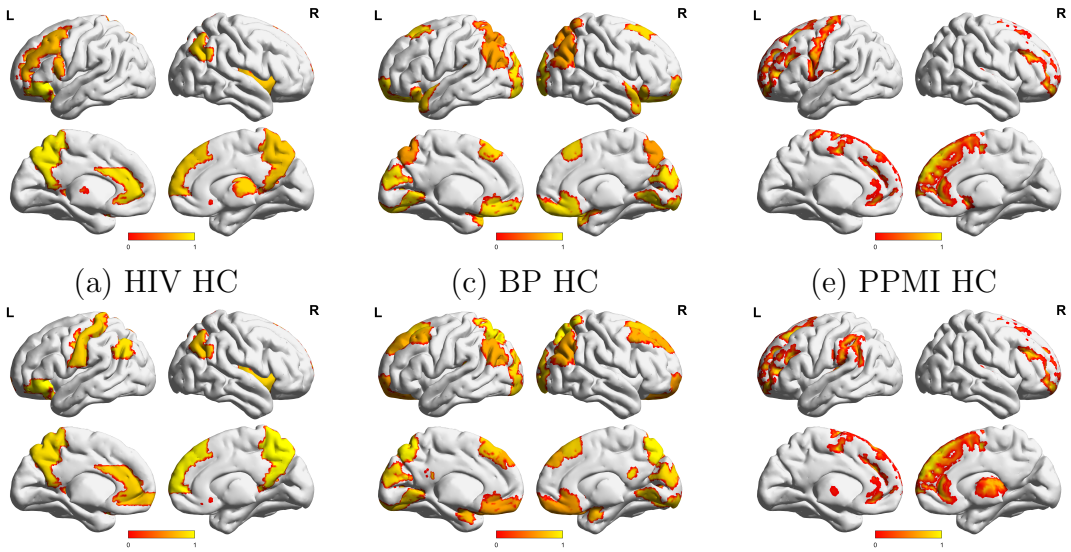
Under a particular parcellation atlas, ROIs on brain networks can be partitioned into neural systems based on their structural and functional roles, making it easier to interpret given explanations from a neuroscience viewpoint. The ROI nodes described on each dataset are mapped into eight commonly used neural systems, including the Visual Network (VN), Auditory Network (AN), Bilateral Limbic Network (BLN), Default Mode Network (DMN), Somato-Motor Network (SMN), Subcortical Network (SN), Memory Network (MN), and Cognitive Control Network (CCN).

Like all graphs, brain networks are composed of nodes and edges, where nodes represent the region of interests (ROIs), and edges represent the correlation between them. We run an explainer on the test datasets and interpret the result from both perspectives. For ROIs, we first apply the edge mask on all samples and sum the edge weights connected to each node to get a node importance level. Then, we rank the importance level and look for the most salient ROIs. For edges, we add all edges in the filtered test dataset, calling it *average graph*. We then use the BrainNet Viewer [68] to plot the average graph and look for the difference in edges.

5.3.1 Salient ROIs

In Figure 5.1, the ROI's average relevance in the given group is represented by the color of the areas. A high score is shown by the bright yellow color, whereas a poor score is indicated by the dark red tint.

The anterior cingulate, paracingulate, and inferior frontal gyri are shown to be important ROIs for HIV disease. This aligns with the study by Ma et al., which states that the regional homogeneity value of the anterior cingulate and paracingulate gyri are decreased [45] in HIV patients. Another study by Li et al. also confirms that lower gray matter volumes are found in the inferior frontal gyrus in HIV patients [40].



(a) HIV HC (c) BP HC (e) PPMI HC

(b) HIV Patient (d) BP Patient (f) PPMI Patient

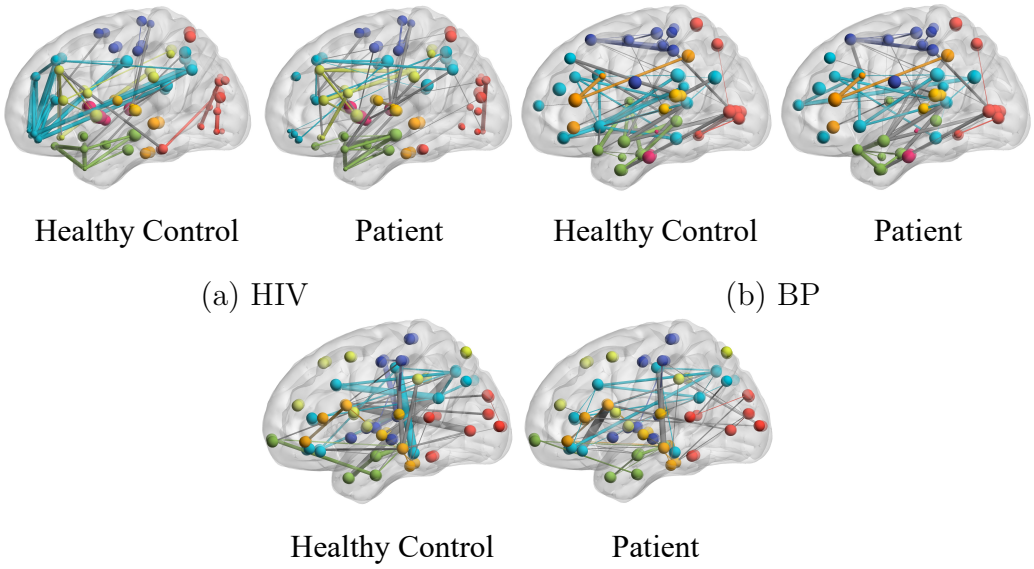
Figure 5.1: Visualization of salient ROIs on the explanation enhanced brain connection network

The individual-level visualizations in Figure 5.1(a)(b) show the difference between HC and HIV patients in those salient ROIs.

For the BP dataset, we observe that the secondary visual cortex and medial to superior temporal gyrus are salient ROIs. We confirm this result with existing research that states BP patients' visual processing functionalities have been affected by the disease [52]. We also confirm the observation with the edge visualizations in Figure 5.1(c)(d).

The rostral middle frontal gyrus and superior frontal gyrus are shown to be important ROIs in the PPMI dataset. The difference can be observed in Figure 5.1(e)(f). The study also confirms that decreased connections in rostral medial frontal gyrus and superior, middle, and inferior frontal gyri are observed in PPMI patients [33].

These observations identify potential biomarkers that could guide further study for the three disorders.



(a) HIV (b) BP

(c) PPMI

Figure 5.2: Visualization of important connections on the explanation enhanced brain connection network.

5.3.2 Edges

The explainer model generates a shared explanation mask $\boldsymbol{M}$. The mask signals connections within brain that are closely related to the disorder. We average the graphs from the test dataset and apply the global mask to obtain an average *explanation graph*. The explanation graph is then filtered so that only top 100 weighted edges remain, calling it G_s'. We compare the explanation subgraphs G_s' of patients with those of healthy controls and identify connections related to specific disorders.

Results are shown in Figure 5.2. Edges connecting nodes within the same neural system (VN, AN, BLN, DMN, SMN, SN, MN, CCN) are colored accordingly. Edges across multiple systems are colored gray. The weight of the edge are shown as the width of the edge.

From the Figure 5.2(a), we observe that the explanation subgraph of HIV patients lacks connections within the DMN system and VN system. These patterns are confirmed by the previous findings that the change in DMN and VN systems, both within-system and inter-system, affects the visual processing difficulties for HIV

patients [27, 20].

In the BP dataset, the healthy controls feature-rich interactions within the BLN community. For the BP patients, however, the connections within that community are much sparser. This observation signals pathological changes in the BLN system. Previous studies support this observation [11, 18]. The studies conclude that the parietal lobe, one of the brain's primary lobes responsible for processing sensory information received from the outside environment and is roughly located in the upper back part of the skull, is mostly linked to Bipolar disorder attacks. The missing links within the BLN system in our image are compatible with current medical knowledge, as parietal lobe ROIs are encompassed in BLN under our parcellation.

Similarly, in the PPMI dataset, Parkinson's patients' experience decreased connections within the SMN system. The SMN system contains primary sensorimotor, premotor, and supplementary motor areas to facilitate voluntary movements. We also confirm this observation with the previous studies, which state that Parkinson's patients experience significant alterations in sensorimotor areas [8]. We also find that Parkinson's patients have sparser connections within the DMN area than those of healthy controls. This is also consistent with the cognition study on Parkinson's patients [59].

Chapter 6

Conclusion

This paper provides a novel interpretable GNN framework for connectome-based brain disease analysis that comprises a brain network-oriented GNN predictor and a globally shared explanation generator. Experiments on real-world neuroscience datasets reveal that our backbone and explanation augmented models have higher prediction ability, and the produced explanation mask validates the disorder-specific interpretations. One limitation of the model might be due to the limited size of neuroimaging datasets. Small datasets limits models' ability to effectively learn the common patterns, which becomes more harmful as the model becomes more complex.

As we tune the GNN models, we find that no current work has been proposed to run a fair comparison between GNN designs for brain networks. Therefore, we present a unified, modular, scalable and reproducible framework for brain network analysis. We tested various combinations of the GNN designs and summarized the best practice for brain network analysis.

As GNNs suffer from over-smoothing and over-squashing problems, we present a transformer-based model, Brain Transformer, and evaluate its performance. The preliminary result proves the predicting power of the transformer model. We also employ differential pooling, which provides potential interpretability and enhanced

performance. A direct future direction is to utilize the attention score and cluster result to interpret the clinical significance of the model's result, including important nodes, edges, and prominent node clusters. Many alterations of the DiffPool layer can be trialed, and clustering-based layers, showing promising results in other areas, can potentially offer a further increase in the performance of the DiffPool layer.

Appendix A

Appendix

A.1 Implementation details

The proposed models discussed in this paper are implemented using PyTorch 1.10.2 [49] and PyTorch Geometric 2.0.3 [19]. A Quadro RTX 8000 GPU with 48GB of memory is used for model training. Hyper-parameters are selected automatically with an open-source AutoML toolkit NNI[1]. Please refer to our repository for comprehensive parameter configurations. The metrics used to evaluate performance are Accuracy, F1 score, and Area Under the ROC Curve (AUC), which are widely used for disease identification. To indicate the robustness of each model, all the reported results are the average performance of ten-fold cross-validation conducted on different train/test splits. The explainer model is available at `https://github.com/DDavid233/BrainNNExplainer_Submission`. The BrainGB project is available at `https://github.com/HennyJie/BrainGB`.

[1] `https://github.com/microsoft/nni/`

A.2 Ethical Statement

The research work is related to human studies. The HIV and Bipolar datasets employed in this study are owned by a third-party organization, where informed consent was obtained for all subjects. The data processed is anonymous with no personally identifiable information. The PPMI dataset is publicly available with restrictions. All studies are conducted according to the Good Clinical Practice guidelines and U.S. 21 CFR Part 50 (Protection of Human Subjects) and under the approval of Institutional Review Boards.

A.3 Collaborations

The IBGNN and BrainGB project is done in collaboration with Hejie Cui, where we made roughly equal contributions. The Brain Transformer project is mainly my own work.

Bibliography

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016. URL `http://manikvarma.org/downloads/XC/XMLRepository.html`.

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

[4] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nat. Rev. Neurosci.*, 10:186–198, 2009.

[5] Chen Cai and Yusu Wang. A Simple Yet Effective Baseline for Non-Attributed Graph Classification. *ArXiv.org*, 2018.

[6] Bokai Cao, Liang Zhan, Xiangnan Kong, S Yu Philip, Nathalie Vizueta, Lori L Altshuler, and Alex D Leow. Identification of discriminative subgraph patterns in fmri brain networks in bipolar affective disorder. In *International Conference on Brain Informatics and Health*, pages 105–114. Springer, 2015.

[7] BJ Casey, Tariq Cannonier, May I Conley, Alexandra O Cohen, Deanna M Barch, Mary M Heitzeg, Mary E Soules, Theresa Teslovich, Danielle V Dellarco,

Hugh Garavan, et al. The adolescent brain cognitive development (abcd) study: imaging acquisition across 21 sites. *Dev Cogn Neurosci*, 32:43–54, 2018.

[8] Julian Caspers, Christian Rubbert, et al. Within-and across-network alterations of the sensorimotor network in parkinson's disease. *Neuroradiology*, 63:2073, 2021.

[9] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. *arXiv preprint arXiv:2004.05718*, 2020.

[10] Hejie Cui, Zijie Lu, Pan Li, and Carl Yang. On positional and structural node features for graph neural networks on non-attributed graphs. *ArXiv.org*, 2021.

[11] Tushar K Das, Jyothika Kumar, Susan Francis, Peter F Liddle, and Lena Palaniyappan. Parietal lobe and disorganisation syndrome in schizophrenia and psychotic bipolar disorder: A bimodal connectivity study. *Psychiatry Research: Neuroimaging*, 303:111139, 2020.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.

[13] Nico U.F. Dosenbach, Damien A. Fair, Alexander L. Cohen, Bradley L. Schlaggar, and Steven E. Petersen. A dual-networks architecture of top-down control. *Trends in Cognitive Sciences*, 12(3):99–105, 2008. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2008.01.001. URL `https://www.sciencedirect.com/science/article/pii/S1364661308000272`.

[14] Chi Thang Duong, Thanh Dat Hoang, Ha The Hien Dang, Quoc Viet Hung Nguyen, and Karl Aberer. On node features for graph neural networks. *ArXiv.org*, 2019.

[15] Martin Dyrba, Michel Grothe, Thomas Kirste, and Stefan J. Teipel. Multimodal Analysis of Functional and Structural Disconnection in Alzheimer's Disease Using Multiple Kernel SVM. *Hum. Brain Mapp.*, 36:2118–2131, 2015.

[16] Farzad V Farahani, Waldemar Karwowski, and Nichole R Lighthall. Application of graph theory for identifying connectivity patterns in human brain networks: a systematic review. *Front. Neurosci.*, 13:585, 2019.

[17] Joshua Faskowitz, Richard F Betzel, and Olaf Sporns. Edges in brain networks: Contributions to models of structure and function. *ArXiv.org*, 2021.

[18] Adele Ferro, Carolina Bonivento, Giuseppe Delvecchio, Marcella Bellani, Cinzia Perlini, Nicola Dusi, Veronica Marinelli, Mirella Ruggeri, A Carlo Altamura, Benedicto Crespo-Facorro, et al. Longitudinal investigation of the parietal lobe anatomy in bipolar disorder and its association with general functioning. *Psychiatry Research: Neuroimaging*, 267:22–31, 2017.

[19] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. In *RLGM@ICLR*, 2019.

[20] Jessica S Flannery, Michael C Riedel, Taylor Salo, Ranjita Poudel, Angela R Laird, Raul Gonzalez, and Matthew T Sutherland. Hiv infection is linked with reduced error-related default mode network suppression and poorer medication management abilities. *medRxiv*, 2021.

[21] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. Large-scale learnable graph convolutional networks. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1416–1424, 2018.

[22] Matthew F. Glasser, Stamatios N. Sotiropoulos, J. Anthony Wilson, Timothy S. Coalson, Bruce Fischl, Jesper L. Andersson, Junqian Xu, Saad Jbabdi, Matthew

Webster, Jonathan R. Polimeni, David C. Van Essen, and Mark Jenkinson. The minimal preprocessing pipelines for the human connectome project. *NeuroImage*, 80:105–124, 2013.

[23] Daniele Grattarola, Daniele Zambon, Filippo Maria Bianchi, and Cesare Alippi. Understanding pooling in graph neural networks. *ArXiv.org*, 2021.

[24] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R. Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *ArXiv.org*, 2021.

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

[26] Lifang He, Kun Chen, Wanwan Xu, Jiayu Zhou, and Fei Wang. Boosted sparse and low-rank tensor regression. In *NeurIPS*, 2018.

[27] Megan M Herting, Kristina A Uban, Paige L Williams, Prapti Gautam, Yanling Huo, Kathleen Malee, Ram Yogev, John Csernansky, Lei Wang, Sharon Nichols, et al. Default mode connectivity in youth with perinatally acquired hiv. *Medicine*, 94, 2015.

[28] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

[29] Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin R. Benson. Combining label propagation and simple models out-performs graph neural networks. *ArXiv.org*, 2020.

[30] Biao Jie, Mingxia Liu, Xi Jiang, and Daoqiang Zhang. Sub-network based kernels for brain network classification. In *ICBC*, 2016.

[31] 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. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2016.09.046. URL `https://www.sciencedirect.com/science/article/pii/S1053811916305237`.

[32] 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.

[33] AT Karagulle Kendi, S Lehericy, et al. Altered diffusion in the frontal lobe in parkinson disease. *American Journal of Neuroradiology*, 29:501, 2008.

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

[35] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34, 2021.

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

[37] Wei Li, Miao Wang, Yapeng Li, Yue Huang, and Xi Chen. A novel brain network construction method for exploring age-related functional reorganization. *Comput. Intell. Neurosci.*, 2016, 2016.

[38] 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. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2021.102233. URL `https://www.sciencedirect.com/science/article/pii/S1361841521002784`.

[39] 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. *Med Image Anal*, 2021.

[40] Yunfang Li et al. Structural gray matter change early in male patients with hiv. *International journal of clinical and experimental medicine*, 7:3362, 2014.

[41] Ye Liu, Lifang He, Bokai Cao, Philip Yu, Ann Ragin, and Alex Leow. Multi-view multi-graph embedding for brain network clustering analysis. In *AAAI*, 2018.

[42] Ye Liu, Lifang He, Bokai Cao, Philip Yu, Ann Ragin, and Alex Leow. Multi-view multi-graph embedding for brain network clustering analysis. In *AAAI*, 2018.

[43] Haiping Lu, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. MPCA: Multilinear Principal Component Analysis of Tensor Objects. *TNN*, 19: 18–39, 2008.

[44] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In *NeurIPS*, 2020.

[45] Qiong Ma, Xiudong Shi, et al. Hiv-associated structural and functional brain alterations in homosexual males. *Frontiers in Neurology*, 12:757374, 2021.

[46] Gustav Martensson, Joana B Pereira, Patrizia Mecocci, Bruno Vellas, Magda Tsolaki, Iwona Kłoszewska, Hilkka Soininen, Simon Lovestone, Andrew Simmons, Giovanni Volpe, et al. Stability of graph theoretical measures in structural brain networks in alzheimer's disease. *Sci. Rep.*, 8:1–15, 2018.

[47] Diego Mesquita, Amauri Souza, and Samuel Kaski. Rethinking pooling in graph neural networks. *NeurIPS*, 2020.

[48] Gowtham Krishnan Murugesan, Chandan Ganesh, Sahil Nalawade, Elizabeth M Davenport, Ben Wagner, Won Hwa Kim, and Joseph A Maldjian. Brainnet: Inference of brain network topology using machine learning. *Brain Connect*, 10: 422–435, 2020.

[49] Adam Paszke, Sam Gross, et al. PyTorch: an imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[50] Jonathan D Power, Alexander L Cohen, Steven M Nelson, Gagan S Wig, Kelly Anne Barnes, Jessica A Church, Alecia C Vogel, Timothy O Laumann, Fran M Miezin, Bradley L Schlaggar, et al. Functional Network Organization of the Human Brain. *Neuron*, 72:665–678, 2011.

[51] Ann B Ragin, Hongyan Du, et al. Structural brain alterations can be detected early in hiv infection. *Neurology*, 79:2328, 2012.

[52] Eric A Reavis, Junghee Lee, et al. Structural and functional connectivity of visual cortex in schizophrenia and bipolar disorder: a graph-theoretic analysis. *Schizophrenia bulletin open*, 1:sgaa056, 2020.

[53] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *ESWC*, 2018.

[54] Weixiang Shao, Lifang He, and Philip S. Yu. Clustering on Multi-source Incomplete Data via Tensor Modeling and Factorization. In *PAKDD*, 2015.

[55] Stephen M Smith. The future of fmri connectivity. *NeuroImage*, 62:1257–1266, 2012.

[56] Stephen M Smith, Peter T Fox, Karla L Miller, David C Glahn, P Mickle Fox, Clare E Mackay, Nicola Filippini, Kate E Watkins, Roberto Toro, Angela R Laird, et al. Correspondence of the brain's functional architecture during activation and rest. *Proc. Natl. Acad. Sci. U.S.A.*, 106:13040–13045, 2009.

[57] Chang Su, Zhenxing Xu, Jyotishman Pathak, and Fei Wang. Deep learning in mental health outcome research: a scoping review. *Transl. Psychiatry*, 10:1–26, 2020.

[58] Heung-Il Suk, Seong-Whan Lee, and Dinggang Shen. Latent feature representation with stacked auto-encoder for ad/mci diagnosis. *Brain Structure and Function*, 220(2):841–859, 2015.

[59] Alessandro Tessitore et al. Default-mode network connectivity in cognitively unimpaired patients with parkinson disease. *Neurology*, 79:2226, 2012.

[60] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. 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. ISSN 1053-8119. doi: https://doi.org/10.1006/nimg.2001.0978. URL `https://www.sciencedirect.com/science/article/pii/S1053811901909784`.

[61] Nathalie Tzourio-Mazoyer, Brigitte Landeau, Dimitri Papathanassiou, Fabrice Crivello, Olivier Etard, Nicolas Delcroix, Bernard Mazoyer, and Marc Joliot. Au-

tomated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *NeuroImage*, 15:273–289, 2002.

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

[64] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.

[65] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep Graph Infomax. In *ICLR*, 2019.

[66] Leanne M Williams. Precision psychiatry: a neural circuit taxonomy for depression and anxiety. *Lancet Psychiatry*, 3:472–480, 2016.

[67] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *AAAI*, volume 33, pages 346–353, 2019.

[68] Mingrui Xia, Jinhui Wang, and Yong He. Brainnet viewer: a network visualization tool for human brain connectomics. *PloS one*, 8:e68910, 2013.

[69] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[70] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[71] Noriaki Yahata, Jun Morimoto, Ryuichiro Hashimoto, Giuseppe Lisi, Kazuhisa Shibata, Yuki Kawakubo, Hitoshi Kuwabara, Miho Kuroda, Takashi Yamada, Fukuda Megumi, et al. A small number of abnormal brain connections predicts adult autism spectrum disorder. *Nat. Commun.*, 7:1–12, 2016.

[72] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34, 2021.

[73] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *arXiv preprint arXiv:1806.08804*, 2018.

[74] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*, 2019.

[75] Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *ICML*, 2019.

[76] Renping Yu, Lishan Qiao, Mingming Chen, Seong-Whan Lee, Xuan Fei, and Dinggang Shen. Weighted graph regularized sparse brain network construction for mci identification. *Pattern Recognit*, 90:220–231, 2019.

[77] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *CoRR*, abs/2012.15445, 2020.

[78] 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. *Front. Aging Neurosci.*, 7:48, 2015.

[79] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1): 1–23, 2019.

[80] Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. *arXiv preprint arXiv:1909.12223*, 2019.

[81] Joelle Zimmermann, John D Griffiths, and Anthony R McIntosh. Unique mapping of structural and functional connectivity on cognition. *J. Neurosci.*, 38: 9658–9667, 2018.