BrainGB: A Benchmark for Brain Network Analysis with Graph Neural Networks

Hejie Cui, Wei Dai, Yanqiao Zhu, Xuan Kan, Antonio Aodong Chen Gu
Joshua Lukemire, Liang Zhan, Lifang He, Ying Guo, Carl Yang

Abstract— Mapping the connectome of the human brain using structural or functional connectivity has become one of the most pervasive paradigms for neuroimaging analysis. Recently, Graph Neural Networks (GNNs) motivated from geometric deep learning have attracted broad interest due to their established power for modeling complex networked data. Despite their superior performance in many fields, there has not yet been a systematic study of how to design effective GNNs for brain network analysis. To bridge this gap, we present BrainGB, a benchmark for brain network analysis with GNNs. BrainGB standardizes the process by (1) summarizing brain network construction pipelines for both functional and structural neuroimaging modalities and (2) modularizing the implementation of GNN designs. We conduct extensive experiments on datasets across cohorts and modalities and recommend a set of general recipes for effective GNN designs on brain networks. To support open and reproducible research on GNN-based brain network analysis, we host the BrainGB website at `https://braingb.us` with models, tutorials, examples, as well as an out-of-box Python package. We hope that this work will provide useful empirical evidence and offer insights for future research in this novel and promising direction.

Index Terms— Brain network analysis, graph neural networks, geometric deep learning for neuroimaging, datasets, benchmarks

I. INTRODUCTION

HUMAN brains are at the center of complex neurobiological systems in which neurons, circuits, and subsystems interact to orchestrate behavior and cognition. Understanding the structures, functions, and mechanisms of human brains has been an intriguing pursuit for researchers with various goals, including neural system simulation, mental disorder therapy, as well as general artificial intelligence. Recent studies in neuroscience and brain imaging have reached the consensus that interactions between brain regions are key driving factors for neural development and disorder analysis [1, 2]. Inspired by graph theory, brain networks composed of nodes and edges are developed to describe the interactions among brain regions.

H. Cui, W. Dai, X. Kan, A. C. Gu, and C. Yang are with the Department of Computer Science, Emory University.

Y. Zhu is with the Department of Computer Science, University of California, Los Angeles.

J. Lukemire and Y. Guo are with the Department of Biostatistics and Bioinformatics, Emory University.

L. Zhan is with the Department of Electrical and Computer Engineering, University of Pittsburgh.

L. He is with the Department of Computer Science and Engineering, Lehigh University.

Correspondence should be addressed to C. Yang (e-mail: j.carlyang@emory.edu).

The human brain can be scanned through various medical imaging techniques, including Magnetic-Resonance Imaging (MRI), Electrogastrography (EGG), Positron Emission Tomography (PET), and so on. Among all these acquisitions, MRI data are the most widely used for brain analysis research. There are also different modalities of MRI data such as functional MRI (fMRI) and Diffusion Tensor Imaging (DTI), from which functional and structural brain networks can be constructed respectively. Specifically, the connectivity in functional brain networks describes correlations between time-series signals of brain regions, while the connectivity in structural brain networks models the physical connectivity between gray matter regions [3]. Both functional and structural connections are widely acknowledged as valuable resources of information for brain investigation [4, 5].

Previous work on brain network analysis has studied shallow models based on graph theory [5, 6] and tensor factorization [7, 8] extensively, which focuses on proposing neurobiologically insightful graph measures and approaches from the node, motif, and graph level to detect network communities or modules and identify central network elements. Methodological developments in graph research enable us to quantify more topological characteristics of complex systems, many of which have already been assessed in brain networks, such as modularity, hierarchy, centrality, and the distribution of network hubs. However, shallow modeling techniques can be inadequate for the sophisticated connectome structures of brain networks [9]. On the other hand, deep learning models have become extraordinarily popular in machine learning, achieving impressive performance on images [10, 11], videos [12], and speech processing tasks [13]. These regular data are represented in 1D/2D/3D Euclidean spaces and can be suitably handled by traditional Recurrent (RNNs) or Convolutional Neural Networks (CNNs). In contrast, the irregular structural and functional brain connectivity networks constructed from neuroimaging data are more complex due to their non-Euclidean characteristics. In recent years, Graph Neural Networks (GNNs) have attracted broad interest due to their established power for analyzing graph-structured data [14–16]. Several pioneering deep models have been devised to predict brain diseases by learning graph structures of brain networks. For instance, Li et al. [1] propose BrainGNN to analyze fMRI data, where ROI-aware graph convolutional layers and ROI-selection pooling layers are designed for neurological biomarker prediction. Kawahara et al. [17] design a CNN framework BrainNetCNN composed of edge-to-edge, edge-to-node, and node-to-graph convolutional filters that leverage the topological locality of structural brain networks. However, they mainly experiment
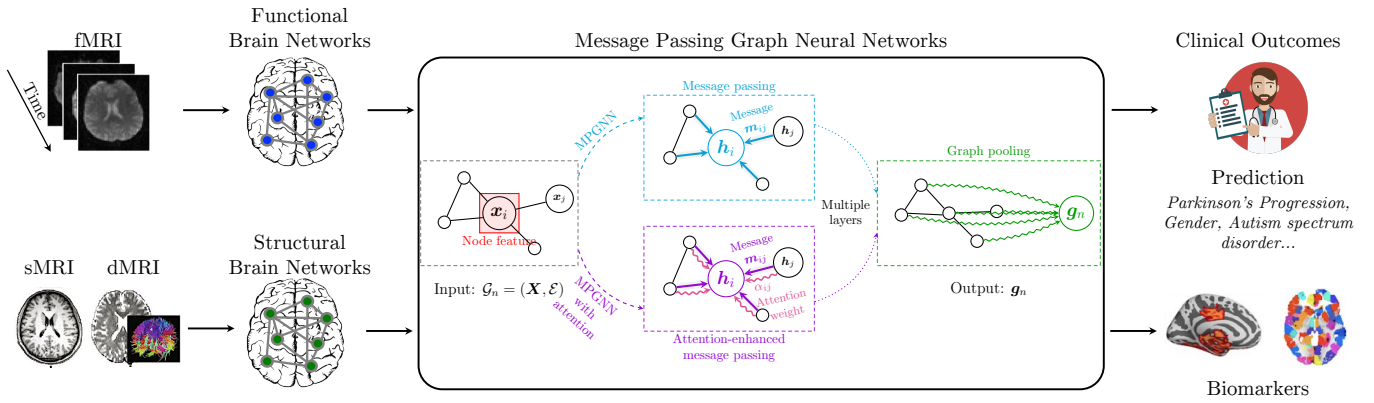
Fig. 1: An overview of our BrainGB framework for brain network analysis with graph neural networks.

with their proposed models on specific private datasets. Due to the ethical issue of human-related research, the datasets used are usually not publicly available and the details of imaging preprocessing are not disclosed, rendering the experiments irreproducible for other researchers.

To address the aforementioned limitations, there is an urgent need for a public benchmark platform to evaluate deep graph models for brain network analysis. However, it is non-trivial to integrate different components within a unified benchmarking platform. Current brain network analyses are typically composed of two steps. The first step is to construct brain networks from neuroimaging data. Then, in the second stage, the resulting brain connectivity between all node pairs is used to classify individuals or predict clinical outcomes. The difficulties in the initial stage are mostly due to restricted data accessibility and sophisticated brain imaging preprocessing and network construction pipelines that differ across cohorts and modalities. The difficulty of the second stage is to establish a standard evaluation pipeline based on fair experimental settings, metrics, and modular-designed baselines that can be easily validated and extended for future research.

In this work, we propose Brain Graph Neural Network Benchmark (BrainGB)—a novel attempt to benchmark brain network analysis with GNNs to the best of our knowledge. The overview of BrainGB is demonstrated in Fig. 1 and the main contributions are four-fold:

- A *unified*, *modular*, *scalable*, and *reproducible* framework is established for brain network analysis with GNNs to facilitate reproducibility. It is designed to enable fair evaluation with accessible datasets, standard settings, and baselines to foster a collaborative environment within computational neuroscience and other related communities.
- We summarize the preprocessing and construction pipelines for both functional and structural brain networks to bridge the gap between the neuroimaging and deep learning community.
- We decompose the design space of interest for GNN-based brain network analysis into four modules: (1) node features, (b) message passing mechanisms, (c) attention mechanisms, and (d) pooling strategies. Different combinations based on these four dimensions are provided

as baselines, and the framework can be easily extended to new variants.
- We conduct a variety of empirical studies and suggest a set of general recipes for effective GNN designs on brain networks, which could be a starting point for further studies.

To foster future research, we release the source code of BrainGB at `https://github.com/HennyJie/Brai nGB` and provide an out-of-box package that can be installed directly, with detailed tutorials available on our hosted website at `https://braingb.us`. Preprocessing instructions and models are provided for standardized model evaluations. We enable the community to collaboratively contribute by submitting their own custom models, and we will maintain a leaderboard to ensure such efforts will be recorded.

II. Preliminaries

A. Brain Network Analysis

Brain networks are complex graphs with anatomic Regions of Interest (ROIs) represented as nodes and connectivities between the ROIs as links [18]. In recent years, the analysis of brain networks has become increasingly important in neuroimaging studies to understand human brain organization across different groups of individuals [19–23]. Abundant findings in neuroscience research suggest that neural circuits are highly related to brain functions, with aberrations in these neural circuits being identified in diseased individuals [24–26].

Formally, in the task of brain network analysis, the input is a brain network dataset $\mathcal{D} = \{\mathcal{G}_n, y_n\}_{n=1}^N$ consisting of N subjects, where $\mathcal{G}_n = \{\mathcal{V}_n, \mathcal{E}_n\}$ represents the brain network of subject n and y_n is the subject's label of the prediction, such as neural diseases. In $\mathcal{D}$, the brain network $\mathcal{G}_n$ of every subject n involves the same set of M nodes defined by the ROIs on a specific brain parcellation, i.e., $\forall n, \mathcal{V}_n = \mathcal{V} = \{v_i\}_{i=1}^M$. The difference across subjects lies in the edge connections $\mathcal{E}_n$ among M brain regions, which are often represented by a weighted adjacency matrix $\boldsymbol{W}_n \in \mathbb{R}^{M \times M}$ describing the connection strengths between ROIs. The edge weights in $\boldsymbol{W}$ are real-valued and the edges are potentially dense and noisy. The model outputs a prediction $\hat{y}_n$ for each subject n, which can be further analyzed in terms of features and biomarkers.

Given brain networks constructed from different modalities such as Diffusion Tensor Imaging (DTI) and functional Magnetic Resonance Imaging (fMRI) [5, 27, 28], effective analysis of the neural connectivities of different label groups (e.g., disease, gender) plays a pivotal role in understanding the biological structures and functions of the complex neural system, which can be helpful in the early diagnosis of neurological disorders and facilitate neuroscience research [29–35]. Previous models on brain networks are mostly shallow, such as graph kernels [36] and tensor factorization [37, 38], which are unable to model the complex graph structures of the brain networks [9].

B. Graph Neural Networks

Graph Neural Networks (GNNs) have revolutionized the field of graph modeling and analysis for real-world networked data such as social networks [14], knowledge graphs [39], protein or gene interaction networks [15], and recommendation systems [40]. The advantage of GNNs is that they can combine node features and graph structures in an end-to-end fashion as needed for specific prediction tasks. A generic framework of GNN could be represented in two phases. In the first phase, it computes the representation h_i of each node $v_i \in \mathcal{V}_n$ by recursively aggregating messages from v_i's multi-hop neighborhood, where h_i^0 is initialized with node features. After getting the last-layer node representation $h^{(L)}$, an extra pooling strategy is adopted to obtain the graph representation. Thereafter, a Multi-Layer Perceptron (MLP) can be applied to make predictions on the downstream tasks.

It is worth noting that brain networks are different from other real-world graphs such as social networks or knowledge graphs, due to (1) the lack of useful initial node (ROI) features on brain networks represented by featureless graphs, (2) the real-valued connection weights that can be both positive or negative, and (3) the ROI identities and their orders are fixed across individual graph samples within the same dataset. The design of GNN models should be customized to fit the unique nature of brain network data. Recently, there have been emerging efforts on GNN-based brain network analysis [1, 17, 41–47]. However, these models are only tested on specific local datasets, mainly due to the convention in neuroscience that researchers are more used to developing methods that are applicable to their specific datasets and the regulatory restrictions that most brain imaging datasets are usually restrictively public, meaning that qualified researchers need to request access to the raw imaging data and preprocess them to obtain brain network data, but they are not allowed to release the preprocessed data afterwards. These challenges largely prohibit the methodology development in computational neuroscience research.

III. BRAIN NETWORK DATASET CONSTRUCTION

A. Background: Diverse Modalities of Brain Imaging

Models of the human brain as a complex network have attracted increasing attention due to their potential for helping understand human cognition and neurological disorders. In practice, human brain data can be acquired through various

scanning techniques [48], such as Magnetic-Resonance Imaging (MRI), Electroencephalography (EEG) and Magnetoencephalography (MEG), Positron Emission Tomography (PET), Single-Photon Emission Computed Tomography (SPECT), and X-ray Computed Tomography (CT). Among them, MRI is one of the most widely used techniques in brain research and clinical practice, due to its large range of available tissue contrast, detailed anatomical visualization, and high sensitivity to abnormalities [49].

1) MRI Data: In this paper, we focus on MRI-derived brain networks. Specifically, for different modalities of MRI data, we can reconstruct different types of brain networks. Functional MRI (fMRI) is one of the most popular modalities for investigating brain function and organization [31, 32, 50] by detecting changes in blood oxygenation and blood flow that occur in response to neural activity. Diffusion-weighted MRI (dMRI), on the other hand, can enable inference about the underlying connection structure in the brain's white matter by recording the diffusion trajectory of molecules (usually water). fMRI focuses on functional activity, while dMRI presents brain structural information from different perspectives. Specifically, two types of brain networks, functional and structural, can be constructed from the aforementioned modalities by following different connectivity generation paradigms [51].

2) Challenges in MRI Preprocessings: The raw MRI data collected from scanners is not directly usable for brain network construction or imaging analysis. A complicated preprocessing pipeline is necessary to remove unwanted artifacts, transform the data into a standard format, and perform structure discovery. Although there are several widely-used neuroimaging data preprocessing tools, such as SPM[1], AFNI[2] and FSL[3], each of them still needs considerable training and learning efforts. Moreover, the functionality of these software varies, and for dMRI, no one software contains all the necessary preprocessing capabilities. In addition, many neuroimaging datasets cannot be made public due to privacy or ethical concerns. Due to the variety of preprocessing approaches and issues with making data publically available, there are difficulties in reproducibility in neuroimaging studies. Additionally, the preprocessing steps are distinctive across modalities. All these challenges make it difficult for deep learning researchers with little knowledge in medical imaging processing to get into the field.

B. Brain Network Construction from Raw Data

In this section, we provide a general overview of the standard preprocessing pipelines for the construction of brain networks of different modalities. Due to the regulation restrictions for direct sharing of the brain network data, we provide two complete pipelines, one for functional brain networks (ABCD[4] specifically) and one for structural brain networks (PPMI[5] specifically), with step-by-step commands and parameter settings on our hosted website for public access[6].

---

[1] https://www.fil.ion.ucl.ac.uk/spm/software/spm12
[2] https://afni.nimh.nih.gov
[3] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSL
[4] https://nda.nih.gov/abcd
[5] https://www.ppmi-info.org
[6] https://braingb.us/preprocessing

Functional MRI Data Preprocessing	SPM 12	AFNI	FSL	Free Surfer	CONN	fMRI Prep	ANTs	Nilearn
Brain Extraction — Remove unnecessary voxels such as bone, air, etc. from T1/T2, apply generated brain mask to fMRI data	✓	✓	✓	✓		✓	✓	✓
Slice-Timing Correction — Adjust for the fact that each slice in the volume is taken at a different time, not all at once	✓	✓	✓	✓	✓	✓		
Motion Correction/Realignment — Correct movement made during scanning by aligning all the functional images with one reference	✓	✓	✓	✓	✓	✓	✓	
Co-registration — Apply EPI distortion correction and align the functional images with the structural images for localization	✓	✓	✓	✓	✓	✓	✓	
Normalization — Warp the data across subjects to a template/atlas standardized space	✓	✓	✓	✓	✓	✓		
Smoothing — Perform weighted averages of individual voxels with neighboring voxels	✓	✓	✓	✓	✓		✓	✓

Functional Brain Network Construction	Recommended Software: **CONN, GraphVar, Brain Connectivity Toolbox**
Brain Region Parcellation — Segment each subject into the ROI defined by the given atlas → **Construct Network** — Calculate pairwise correlations between ROIs as edges	

Fig. 2: The framework of fMRI data preprocessing and functional brain network construction procedures, with recommended tools for each step shown on the right. The more commonly-used tools for the functional modality are placed at the front.

Diffusion MRI Data Preprocessing	FSL	AFNI	Free Surfer	Track Vis	3D Slider	Tortoise	MRtrix3	DSI Studio	DIPY	Tracto Flow
Eddy-current and Head Motion Correction — Align all raw images to the b0 image to correct for head motion and eddy current distortions	✓	✓	✓		✓	✓	✓	✓	✓	✓
EPI Induced Susceptibility Artifacts Correction — Correct the spatially nonlinear distortions caused by B_0 inhomogeneities in Echo-planar imaging	✓	✓	✓		✓	✓				✓
Brain Extraction — Remove voxels not necessary for analysis such as bone, dura, air, etc., leaving just the brain	✓	✓	✓		✓		✓	✓	✓	✓
Reconstruct Local Diffusion Pattern — Fit a diffusion tensor model at each voxel on preprocessed and eddy current corrected data	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tractography — Reconstruct brain connectivity graphs using whole brain tractography algorithms like FACT	✓	✓		✓	✓		✓	✓	✓	✓
Brain Region Parcellation — Parcellate ROIs from T1-weighted structural MRI and map those ROIs to DTI space	✓	✓				✓	✓	✓		✓

Structural Brain Network Construction	Recommended Software: **FSL, Metric, DSI Studio**
Construct Network — Compute the network based on the generated label and the reconstructed whole brain tractography	

Fig. 3: The framework of dMRI data preprocessing and structural brain network construction procedures, with recommended tools for each step shown on the right. The more commonly-used tools for the structural modality are placed at the front.

1) Functional Brain Network Construction: The left side of Fig. 2 shows a standard preprocessing procedure for functional brain imaging, with the corresponding commonly-used toolboxes (i.e., SPM12[1], AFNI[2], FSL[3], FreeSurfer[7], CONN[8], fMRI Prep[9], ANTs[10], Nilearn[11]) shown on the right side. Note that each step in the preprocessing and network construction pipeline needs quality control by the experts, and the specific order of preprocessing steps may change slightly based on the acquisition conditions of the dataset. Some representative functional neuroimaging datasets in literature to facilitate scientific research include ADHD 200 [52], ADNI (fMRI part) [53], HCP 900 [54], ABIDE [55], etc.

To measure functional connectivity, some preprocessing of the fMRI time series is often performed including detrending, demeaning, and whitening fMRI BOLD time series at each voxel [56]. To construct the brain networks, a brain atlas or a set of Regions of Interest (ROI) are selected to define the nodes. Then, the representative fMRI BOLD series from each node are obtained by either averaging or performing Singular Value Decomposition (SVD) on the time series from all the voxels within the node. Various measures have been proposed for assessing brain connectivity between pairs of nodes. One of the simplest and most frequently used methods in the neuroimaging community is via pairwise correlations between BOLD time courses from two ROIs. Other methods include partial correlations [56], mutual information, coherence, Granger causality [57]. After selecting the Functional Connectivity (FC) measure, one can evaluate the strength of connectivity between each pair of ROIs. Often, some transformation, such as the Fisher's transformation, is performed to transform the original FC measures to improve their distribution properties. The transformed FC measures can then be utilized for the subsequent analysis of functional brain networks.

To facilitate public testing, we take Adolescent Brain Cognitive Development Study (ABCD) as an example and

---

[7] https://surfer.nmr.mgh.harvard.edu
[8] https://web.conn-toolbox.org/home
[9] https://fmriprep.org/en/stable/index.html
[10] http://stnava.github.io/ANTs
[11] https://nilearn.github.io/stable/index.html

provide a step-by-step instruction for functional brain network construction on our hosted BrainGB website[6]. The ABCD-HCP BIDS[12] pipeline is used to preprocess the data. In brief, anatomical preprocessing included normalization, co-registration, segmentation, and brain extraction. Functional data preprocessing included slice-time correction, motion correction, distortion correction, co-registration, normalization, and spatial smoothing. Brain parcellation schemes were then applied to the functional data to obtain time courses for each ROI, and Pearson correlation was used to construct brain networks representing the connectivity between ROIs.

2) Structural Brain Network Construction: Structural brain networks provide a systematic perspective for studying the anatomical and physiological organization of human brains and help to understand how brain structure influences function. Some representative neuroimaging studies include diffusion MRI data are PPMI [58], ADNI [53], HCP [54], AIBL [59], OASIS [59], etc. The commonly-used toolboxes for dMRI include FSL[3], AFNI[2], FreeSurfer[7], TrackVis[13], 3D Slicer[14], Tortoise[15], MRtrix3[16], DSI Studio[17].

The left side of Fig. 3 summarizes the pipeline for reconstructing the structural brain network. Preprocessing steps for the dMRI data include removal of eddy current-induced distortions, brain extraction, and co-registration between diffusion and structural images. Next, some modeling strategies are applied to reconstruct the local diffusion patterns. Commonly adopted models include the DTI modeling, which fits a tensor model or muti-tensor model [60] to capture the local diffusion patterns, and the Ball and Sticks model [61]. After reconstructing the local diffusion patterns, a tractography algorithm is performed to computationally reconstruct fiber tract connections between brain regions. Commonly-used algorithms include the deterministic tractography [62] and the probabilistic tractography [63]. The deterministic tractography connects neighboring voxels from seed regions based on the major direction of the DTI tensor. The probabilistic tractography involves first estimating fiber orientation and its uncertainty at each voxel and building a diffusion path probability map based on the estimated orientation and uncertainty. While deterministic tractography is a more computationally efficient approach to reconstruct major fiber bundles in the brain, probabilistic tractography has become more popular because it is more robust to noise and allows tractography to progress beyond uncertain regions by taking into account uncertainty in fiber orientations at each voxel [64]. To construct the structural network, the structure connectivity for each node pair is calculated based on the empirical probability of fiber tracts connecting the two regions. Note that each step of network construction ideally needs quality control from experts.

Similarly to functional brain network construction, we take PPMI as an example and provide an instruction pipeline for structural brain network construction on our hosted BrainGB

website[6]. Specifically, the Diffusion Toolkit from TrackVis is used to reconstruct local diffusion patterns and tractography. The brain region parcellation is completed with both FSL and Freesurfer. Then local diffusion pattern reconstruction and the network computation are further performed by calculating the number of fibers within each ROI after removing the false positive ones.

C. Discussions

In addition to the mainstream methods of constructing connections in brain networks discussed above, there are also other ways to construct different types of edges. For example, directional connectivity that characterizes effective interactions for fMRI [65]; hybrid functional brain networks where different orders of relationships can be sensitive to different levels of signal changes [66]; and dynamic functional brain networks which include derivatives of windowed functional network connectivity in the identification of reoccurring states of connectivity [65, 67]. Apart from fMRI and DTI, the most commonly used modalities to construct functional and structural brain networks, other neuroimaging modalities have also been explored in literature, such as metabolic brain network constructed from PET imaging [68], functional brain network constructed from EEG signals [69], etc. Recent studies have shown that the combination of both functional and structural neuroimaging modalities can be more effective than using only a single one, which can exploit complementary information across different modalities [4, 70].

IV. GNN BASELINES FOR BRAIN NETWORK ANALYSIS

The process of applying GNNs to brain networks starts from initialization of the ROI features, followed by the forward pass which includes two phases, message passing, and pooling. The learned graph-level representation then can be utilized for brain disease analysis. In the machine learning domain, the rapid evolution of GNNs has led to a growing number of new architectures. Specifically for GNNs on brain network analysis, we decompose the design space of interest for basic message passing GNNs into four modules: node feature construction, message passing, attention enhanced message passing, and pooling strategies. An illustration of these modules is shown in the middle of Fig. 1.

A. Node Feature Construction

In neuroscience analysis, researchers mostly focus on brain connectivity represented by a featureless graph. To apply GNNs on non-attributed brain networks, researchers in the graph machine learning domain have studied several practical methods to initialize node features [71, 72]. In this paper, we focus on the following node features that can be categorized as positional or structural:

- *Identity*: A unique one-hot feature vector is initialized for each node [73, 74]. By giving each ROI in the brain network a unique high-dimensional vector, this identity node feature allows the GNN model to learn the relative positions of the nodes by memorizing their

---

k-hop neighbors. They are essentially the same as random initialization considering the parameters in the first linear layer of the GNN are randomly initialized.

- *Eigen*: Eigen decomposition is performed on the weighted matrix describing the connection strengths between ROIs and then the top k eigenvectors are used to generate a k-dimensional feature vector for each node [75–77]. The optimal value of k is decided by grid search. This feature is essentially dimension reduction and targets at grouping brain regions with respect to their positions, with global graph information condensed into a low-dimensional representation.

- *Degree*: The degree value of each node is obtained as a one-dimensional vector as the node feature. This feature captures structural information of brain regions, meaning that neighborhood structural similarity of two regions will be partially recorded in the initialized node features.

- *Degree profile*: This method takes advantages of existing local statistical measures on degree profiles [78], where each feature $\boldsymbol{x}_i$ of node v_i on graph $\mathcal{G}_n$ is computed as

$$\boldsymbol{x}_i = [\deg(v_i) \parallel \min(\mathcal{D}_i) \parallel \max(\mathcal{D}_i) \\ \parallel \text{mean}(\mathcal{D}_i) \parallel \text{std}(\mathcal{D}_i)], \quad (1)$$

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

- *Connection profile*: The corresponding row for each node in the edge weight matrix is utilized as the initial node feature, which contains connections with respect to all other nodes in the brain network. This feature aligns with the common practice of using pairwise connections to perform brain parcellation. Also, it reflects the whole picture of connection information in the brain network.

B. Message Passing Mechanisms

The power of most GNNs to learn structures lies in their message passing schemes, where the node representation is updated iteratively by aggregating neighbor features through local connections. In each layer l, the node representation $\boldsymbol{h}_i^l$ is updated through two steps, namely message passing and update respectively. In the message passing step (Eq. 2), each node v_i receives messages from all its neighbors, and then all the messages are aggregated with a sum function:

$$\boldsymbol{m}_i^l = \sum_{j \in \mathcal{N}_i} \boldsymbol{m}_{ij} = \sum_{j \in \mathcal{N}_i} M_l\left(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, w_{ij}\right), \quad (2)$$

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 the update step (Eq. 3), the embedding of each node is updated based on the aggregated messages from Eq. 2 and optionally the previous embedding of node v_i, where the update function can be arbitrary differentiable functions (e.g., concat the aggregated message with the previous node embedding and then pass them into a learnable linear layer).

$$\boldsymbol{h}_i^{l+1} = U_l\left(\boldsymbol{h}_i^l, \boldsymbol{m}_i^l\right), \quad (3)$$

where U_l stands for the update function and the number of running steps L is defined by the number of GNN layers. The

message passing mechanism can leverage both permutation equivariance and inductive bias towards learning local structures and achieve good generalization on new graphs. For brain networks, whether incorporating connections into the message function is beneficial for graph-level prediction tasks remains to be investigated. In this paper, we discuss the influence of different message function M_l designs including:

- *Edge weighted*: The message $\boldsymbol{m}_{ij}$ passed from node v_j to node v_i is calculated as the representation of node v_j weighted by the corresponding edge weight w_{ij}, that is

$$\boldsymbol{m}_{ij} = \boldsymbol{h}_j \cdot w_{ij}. \quad (4)$$

This is the standard message passing implementation in Graph Convolutional Network (GCN) [14] when $w_{ij} = 1/N_i$. With this message vector design, the update of each brain region representation is influenced by its neighbor regions weighted by the connection strength between them.

- *Bin concat*: In this scheme, we map the edge w_{ij} into one of the equally split T buckets based on its weight value. Each bucket corresponds to a learnable representation $\boldsymbol{b}_t$, $t = \{1 \ldots T\}$. The total bucket number encompassing the entire value range of edge weights is determined by grid search and the representation dimension of each bin is set to the same as node features. Specifically, given the number of buckets is T, we first rank all the edge weights and then divide them into the equally divided T buckets from the lowest to the highest. All edges in the same bucket will be mapped to the same learnable vector $\boldsymbol{b}_t$, so region connections with similar strength are binned together. In our experiment, we simply select from [5, 10, 15, 20] as the possible number of buckets for grid search, which is a common practice in machine learning for hyperparameter tuning. The message $\boldsymbol{m}_j$ passed from node v_j to node v_i is calculated as the concatenation of the representation of node v_j and its corresponding bucket representation $\boldsymbol{b}_t$ followed by an MLP,

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_j \parallel \boldsymbol{b}_t). \quad (5)$$

The usage of bins helps to clusters region connections with similar strengths. By concatenating with the unique neighbor node representation, this message captures both common and peculiar characteristics of each neighbor.

- *Edge weight concat*: The message $\boldsymbol{m}_{ij}$ passed from node v_j to node v_i is represented as the concatenation of the representation of node v_j and the scaled edge weight $d \cdot w_{ij}$, followed by a MLP,

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_j \parallel d \cdot w_{ij}), \quad (6)$$

where d is a constant equals to the dimension number of node features. The motivation behind edge weight scaling is to increase the influence of edge features to the same scale as node features. Compared with bin concat where edges with weight values in the same bin interval share the same initial edge representation, directly concatenating the scaled edge weights as the edge representations can retain the original edge information, therefore reserving more uniqueness on the pairwise connection when performing the aggregation from neighboring brain regions.

- *Node edge concat*: To investigate the influence of preserving the brain region representation from the last time step while iterative updating the new representation, we design a message $\boldsymbol{m}_j$ as the concatenation of both embeddings of node v_i, v_i and the edge weight w_{ij} between them, followed by a MLP, that is

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_i \parallel \boldsymbol{h}_j \parallel w_{ij}). \tag{7}$$

In this paradigm, every message passed from the local neighbors of each central node is reinforced with its representation from the last time step. This design may alleviate the over-smoothing problem of GNNs, where the feature distance between all nodes becomes too close and not distinguishable after layers of convolutions.

- *Node concat*: Since the effect of involving connection weights into message passing is still unknown, we also include another message $\boldsymbol{m}_{ij}$ similar to *node edge concat* but without the concatenation of edge weights, where

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_i \parallel \boldsymbol{h}_j). \tag{8}$$

C. Attention-Enhanced Message Passing

Attention is arguably one of the most important mechanisms in modern deep learning [79, 80]. It is inspired by human cognitive systems that tend to selectively concentrate on the important parts as needed when processing large amounts of information. Various fields in deep learning communities such as natural language processing [81] and computer vision [82] have widely benefited from attention mechanisms in terms of model efficiency and accuracy. The attention mechanism can also be used to enhance the message passing scheme of GNNs, while also providing interpretations over the edge importance.

Specifically in brain network analysis, by utilizing the attention-enhanced version of message passing, the model updates the brain region representation in a data-driven way, where adjustable attention weights from each local neighbor perform as an additional influence factor besides the neural signals represented by edge weights. It is worth noting that the traditional designs of graph attention mechanisms on general graphs usually do not take the edge attributes (i.e., connection weights in the brain network scenario) into consideration. However, for brain networks, the correlation between two regions contains meaningful biomedical information and might be helpful for graph-level tasks. In this paper, we design several attention-enhanced message passing mechanisms including:

- *Attention weighted*: This is the original GAT [16] on general graphs without involving edge attributes. The message from node v_j to v_i is weighted by the corresponding attention score α_{ij} as

$$\boldsymbol{m}_{ij} = \boldsymbol{h}_j \cdot \alpha_{ij}. \tag{9}$$

The α_{ij} is calculated from a single-layer feed-forward neural network parameterized by a weight vector $\mathbf{a}$, followed by the LeakyReLU nonlinearity σ,

$$\alpha_{ij} = \frac{\exp\left(\sigma\left(\boldsymbol{a}^\top [\boldsymbol{\Theta} \boldsymbol{x}_i \parallel \boldsymbol{\Theta} \boldsymbol{x}_j]\right)\right)}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp\left(\sigma\left(\boldsymbol{a}^\top [\boldsymbol{\Theta} \boldsymbol{x}_i \parallel \boldsymbol{\Theta} \boldsymbol{x}_k]\right)\right)}, \tag{10}$$

where $\boldsymbol{\Theta}$ is a learnable linear transformation matrix.

- *Edge weighted w/ attn*: This is the attention-enhanced version of *edge weighted* message passing in Eq. 4. The message from v_j to v_i is obtained as the multiplication of node v_j's representation $\boldsymbol{h}_j$, the edge weight w_{ij} and the attention score α_{ij} in Eq. 10,

$$\boldsymbol{m}_{ij} = \boldsymbol{h}_j \cdot \alpha_{ij} \cdot w_{ij}. \tag{11}$$

- *Attention edge sum*: This is another version of attention-enhanced *edge weighted* (Eq. 4) message passing. The edge weight w_{ij} and the attention score α_{ij} are first summed, then used as the impact factor on the node embedding $\boldsymbol{h}_j$,

$$\boldsymbol{m}_{ij} = \boldsymbol{h}_j \cdot (\alpha_{ij} + w_{ij}). \tag{12}$$

- *Node edge concat w/ attn*: This is the attention-enhanced version of *node edge concat* (Eq. 7) message passing, where the attention score α_{ij} (Eq. 10) between node v_i and v_j is multiplied on the node representation $\boldsymbol{h}_j$ before concatenation, followed by a MLP,

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_i \parallel (\boldsymbol{h}_j \cdot \alpha_{ij}) \parallel w_{ij}). \tag{13}$$

- *Node concat w/ attn*: This design corresponds to the attention-enhanced version of *node concat* (Eq. 8) message passing, where the attention score α_{ij} (Eq. 10) between node v_i and node v_j is multiplied on the node representation $\boldsymbol{h}_j$ before concatenation, followed by a MLP,

$$\boldsymbol{m}_{ij} = \text{MLP}(\boldsymbol{h}_i \parallel (\boldsymbol{h}_j \cdot \alpha_{ij})). \tag{14}$$

D. Pooling Strategies

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

$$\boldsymbol{g}_n = R\left(\{\boldsymbol{h}_k \mid v_k \in \mathcal{G}_n\}\right). \tag{15}$$

The pooling function R operates on the set of node vectors and is invariant to permutations of the node vectors. In this paper, we cover three basic global pooling operators [83, 84]:

- *Mean pooling*: The graph-level representation is obtained by averaging node features. For each single graph $\mathcal{G}_n$, the graph-level representation is computed as

$$\boldsymbol{g}_n = \frac{1}{M} \sum_{k=1}^{M} \boldsymbol{h}_k. \tag{16}$$

- *Sum pooling*: The graph-level representation is obtained by summing up all node features. For each single graph $\mathcal{G}_n$, the graph-level representation is computed as

$$\boldsymbol{g}_n = \sum_{k=1}^{M} \boldsymbol{h}_k. \tag{17}$$

- *Concat pooling*: The graph-level representation is obtained by concatenating node features of all nodes contained in the graph. For each single graph $\mathcal{G}_n$, the graph-level representation is computed as

$$\boldsymbol{g}_n = \parallel_{k=1}^{M} \boldsymbol{h}_k = \boldsymbol{h}_1 \parallel \boldsymbol{h}_2 \parallel \ldots \parallel \boldsymbol{h}_M. \tag{18}$$

Note that there are also other complex pooling strategies such as hierarchical pooling [85], learnable pooling [86] and clustering readout [87], which are usually viewed as independent GNN architecture designs that are not defined based on combinative modules. Here we include the representative method of DiffPool [85] to provide a view of the comparison between basic and more complex pooling methods.

V. EXPERIMENTAL ANALYSIS AND INSIGHTS

In this section, we show experimental results on brain networks generated from real-world neuroimaging studies with different GNN modular designs. Varying each design dimension under each module results in a total of 375 different architectures. Note that here we do not aim to cover all combinations, but to quickly find a relatively good one. Furthermore, we emphasize that the design space can be expanded as new design dimensions emerge.

A. Experimental Settings

1) Datasets: To establish a benchmark for generic brain network analysis models, we include four datasets processed and constructed from different neuroimaging modalities, specifically fMRI (HIV [38], PNC[18], ABCD[4]) and dMRI (PPMI[5]), based on different brain atlas. For the HIV and PPMI datasets, the task is to classify patients from healthy control (Patient, Normal Control); while for the PNC and ABCD datasets, the task is gender prediction (Male, Female). We intentionally cover such a diverse set of datasets from different modalities (and preprocessing procedures/parcellations/tasks), because our purpose is to establish a benchmark for generic brain network analysis models. Thus observations on a diverse set of datasets can be more instructive for methodology focused studies. All the datasets we used have been visually checked by imaging experts in our team for quality control. Among these four datasets, PNC, PPMI, and ABCD are restrictively publicly available ones that can be requested and downloaded from their official website. The dataset information is summarized in TABLE I. Since the datasets can be acquired from multiple sites, multisite issues need to be addressed when performing the analysis on the constructed networks. Over the past few years, ComBat techniques [88, 89] from the microarray literature have started to be used more frequently to deal with multi-site batch effects. Since our benchmark focuses more on a comprehensive overview of brain network construction and effective GNN designs for brain networks, advanced methods for handling multi-site issues are out of the scope of this work. Interested readers can refer to [90–94] for more advanced multisite data handling methods.

- *Human Immunodeficiency Virus Infection (HIV):* This dataset is collected from the Chicago Early HIV Infection Study at Northwestern University [95]. The clinical cohort includes fMRI imaging of 70 subjects, 35 of which are early HIV patients and the other 35 are seronegative controls. The preprocessing includes realignment to the

first volume, followed by slice timing correction, normalization, and spatial smoothness, band-pass filtering, and linear trend removal of the time series. We focus on the 116 anatomical ROIs [96] and extract a sequence of time courses from them. Finally, brain networks with 90 cerebral regions are constructed, with links representing the correlations between ROIs.

- *Philadelphia Neuroimaging Cohort (PNC):* This rs-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. [97]. The preprocessing includes slice timing correction, motion correction, registration, normalization, 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 [98].

- *Parkinson's Progression Markers Initiative (PPMI):* This dataset is from a collaborative study for Parkinson's Research to improve PD therapeutics. We consider the DTI acquisition of 754 subjects, with 596 Parkinson's disease patients and 158 healthy controls. The raw data are first aligned to correct for head motion and eddy current distortions. Then the non-brain tissue is removed and the skull-stripped images are linearly aligned and registered. 84 ROIs are parcellated from T1-weighted structural MRI based on the Desikan-Killiany' cortical atlas [99] and the brain network is reconstructed using the deterministic 2nd-order Runge-Kutta (RK2) whole-brain tractography algorithm [64].

- *Adolescent Brain Cognitive Development Study (ABCD):* This study recruits children aged 9-10 years across 21 sites in the U.S. Each child is followed into early adulthood, with repeated imaging scans, as well as extensive psychological and cognitive tests [100]. After selection, 7,901 children are included in the analysis, with 3,961 (50.1%) female. We use rs-fMRI scans for the baseline visit processed with the standard and open-source ABCD-HCP BIDS fMRI Pipeline[12]. 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 [101].

Structural connectivity and functional connectivity are different in their strength and sparsity, thus need to be handled differently. For structural connectivity, we normalize the edge weights by dividing each value by the maximum value in a sample. The processed edge weights are thus ranged from 0 to 1. For functional connectivity, we follow common practice to remove the negative values for GNNs that cannot handle negative values (like GCN), and keep them for GNNs that can handle negative values (like GAT).

2) Baselines: For comprehensiveness, we compare our modular design with competitors of both shallow and deep models. The shallow methods we consider include M2E [7],

---

TABLE I: Dataset summarization.

Dataset	Modality	# Samples	Atlas	Size	Response	# Classes
HIV	fMRI	70	AAL 116	90×90	Disease	2
PNC	fMRI	503	Power 264	232×232	Gender	2
PPMI	DTI	754	Desikan-Killiany	84×84	Disease	2
ABCD	fMRI	7,901	HCP 360	360×360	Gender	2

TABLE II: Performance report (%) of different message passing GNNs in the four-modular design space with other two representative baselines on four datasets. We highlight the best performed one in each module based on AUC, since it is not sensitive to the changes in the class distribution, providing a fair evaluation on unbalanced datasets like PPMI.

Module	Method	HIV			PNC			PPMI			ABCD		
		Accuracy	F1	AUC	Accuracy	F1	AUC	Accuracy	F1	AUC	Accuracy	F1	AUC
Node Features	*Identity*	50.00±0.00	33.33±0.00	46.73±10.57	57.34±0.17	36.44±0.17	52.58±4.80	79.25±0.24	44.21±0.08	59.65±6.80	49.97±0.13	33.32±0.06	50.00±0.20
	Eigen	65.71±2.86	65.45±2.69	65.31±2.89	51.40±3.92	48.63±5.42	50.18±7.57	74.09±2.77	47.36±4.26	49.21±1.58	50.79±0.82	50.79±0.83	51.18±1.16
	Degree	44.29±5.35	35.50±6.10	42.04±4.00	63.89±2.27	59.69±3.85	70.25±4.38	79.52±2.31	49.40±5.17	59.73±4.31	63.46±1.29	63.45±1.28	68.16±1.41
	Degree profile	50.00±0.00	33.33±0.00	50.00±0.00	51.40±7.21	33.80±3.21	50.00±0.00	77.02±1.97	49.45±3.51	58.65±2.44	49.92±0.11	33.30±0.05	50.00±0.00
	Connection profile	65.71±13.85	64.11±13.99	**75.10±16.95**	69.83±4.15	66.20±4.74	**76.69±5.04**	77.99±2.78	52.96±4.52	**65.77±4.09**	82.42±1.93	82.30±2.08	**91.33±0.77**
Message Passing	*Edge weighted*	50.00±0.00	33.33±0.00	49.80±4.20	64.87±5.44	59.70±7.04	69.98±4.19	79.25±0.24	44.21±0.08	62.26±2.80	74.47±1.17	74.36±1.23	82.37±1.46
	Bin concat	50.00±0.00	33.33±0.00	49.39±9.25	54.74±5.88	36.42±3.97	61.68±3.91	79.25±0.24	44.21±0.08	52.67±7.16	53.72±4.97	43.26±12.43	61.86±5.79
	Edge weight concat	51.43±2.86	34.36±6.88	48.16±10.13	63.68±3.31	60.27±5.97	67.34±3.02	79.25±0.24	44.21±0.08	59.72±4.65	64.59±1.30	64.30±1.43	70.63±1.02
	Node edge concat	65.71±13.85	64.11±13.99	75.10±16.95	69.83±4.15	66.20±4.74	76.69±5.04	77.99±2.78	52.96±4.52	65.77±4.09	82.42±1.93	82.30±2.08	91.33±0.77
	Node concat	70.00±15.91	68.83±17.57	**77.96±8.20**	70.63±2.35	67.12±1.81	**78.32±1.42**	78.41±1.62	54.46±3.08	**68.34±1.89**	80.50±2.27	80.10±2.47	**91.36±0.92**
Message Passing w/ Attention	*Attention weighted*	50.00±0.00	33.33±0.00	49.80±8.52	65.09±2.21	60.74±4.89	69.79±4.24	79.25±0.24	44.21±0.08	63.24±3.77	77.74±0.97	77.70±1.01	85.10±1.10
	Edge weighted w/ attn	50.00±0.00	33.33±0.00	42.04±15.63	62.90±1.22	61.14±0.57	69.74±2.37	79.25±0.24	44.21±0.08	54.92±4.80	78.04±1.96	77.81±2.33	86.86±0.63
	Attention edge sum	51.43±7.00	49.13±5.65	54.49±15.67	61.51±2.86	55.36±4.76	69.38±3.50	79.11±0.40	44.17±0.12	60.47±6.26	75.71±1.52	75.59±1.68	83.78±0.82
	Node edge concat w/ attn	72.86±11.43	72.52±11.72	78.37±10.85	67.66±5.07	64.69±5.36	74.52±1.20	77.30±1.52	50.96±4.20	63.93±4.89	83.10±0.47	83.03±0.52	**91.85±0.29**
	Node concat w/ attn	71.43±9.04	70.47±9.26	**82.04±11.21**	68.85±6.42	64.29±10.15	**75.36±5.09**	78.41±1.43	49.98±1.87	**68.14±5.01**	83.19±0.93	83.12±0.96	91.55±0.59
Pooling Strategies	*Mean pooling*	47.14±15.39	41.71±17.36	58.78±18.63	66.86±2.33	61.39±4.88	74.20±3.39	79.25±0.24	44.21±0.08	59.64±5.47	81.13±0.35	81.06±0.34	88.49±1.12
	Sum pooling	57.14±9.04	52.23±12.65	57.96±11.15	60.13±2.87	53.96±7.61	66.11±4.22	79.39±0.52	47.68±3.12	61.29±2.11	77.48±3.75	76.96±4.58	87.90±0.65
	Concat pooling	65.71±13.85	64.11±13.99	75.10±16.95	69.83±4.15	66.20±4.74	**76.69±5.04**	77.99±2.78	52.96±4.52	**65.77±4.09**	82.42±1.93	82.30±2.08	**91.33±0.77**
	DiffPool	72.86±21.19	70.22±23.91	**76.57±17.16**	62.72±12.40	75.95±4.28	64.08±16.71	78.42±3.53	56.55±6.48	63.07±7.77	76.45±1.44	76.35±1.52	83.92±1.25
Shallow Baselines	M2E	57.14±19.17	53.71±19.80	57.50±18.71	53.76±4.94	46.10±6.94	49.70±5.18	78.69±1.78	45.81±4.17	50.39±2.59	50.10±1.90	49.95±1.88	50.10±1.90
	MPCA	67.14±20.25	64.28±23.47	69.17±20.17	76.76±4.30	75.95±4.28	77.59±5.00	79.15±0.57	44.18±0.18	50.00±0.00	88.94±1.64	88.94±1.64	88.94±1.64
	MK-SVM	65.71±7.00	62.08±7.49	65.83±7.41	78.38±5.09	77.55±5.83	77.57±5.65	79.15±0.57	44.18±0.18	50.00±0.00	89.42±0.97	89.42±0.97	89.42±0.97
Deep Baselines	BrainNetCNN	60.21±17.16	60.12±13.56	70.93±4.01	71.93±4.90	69.94±5.42	78.50±3.28	77.24±2.09	50.24±3.09	58.76±8.95	85.1±0.92	85.7±0.83	93.5±0.34
	BrainGNN	62.98±11.15	60.45±8.96	68.03±9.16	70.62±4.85	68.93±4.01	77.53±3.23	79.17±1.22	44.19±3.11	45.26±3.65	OOM	OOM	OOM

MPCA [102], and MK-SVM [103], where the output graph-level embeddings are evaluated using logistic regression classifiers. Specifically, M2E is a partially-symmetric tensor factorization based method for brain network analysis, and it has been empirically compared with spectral embedding clustering methods such as SEC [104] or spectral learning frameworks such as AMGL [105]; MPCA is proposed for the feature extraction and analysis of tensor objects such as neuroimaging; multiple kernel SVM (MK-SVM) is essentially an extension of the conventional SVM algorithm and has been applied for the analysis of functional and structural connectivity in Alzheimer's disease. We also include two state-of-the-art deep models specifically designed for brain networks: BrainGNN [1] and BrainNetCNN [17]. The message passing in BrainGNN is Edge weighted and it further leverages additional regional information (such as coordinates or ROI ordering based one-hot embeddings) to assign a separate GCN kernel for each ROI where ROIs in the same community are embedded by the similar kernel and those in different communities are embedded in different ways, but this will introduce a lot of additional model parameters and make the model hard to train. On the other hand, BrainNetCNN models the adjacency matrix of a brain network as a 2D image and does not follow the message passing mechanism as we discussed in Section IV-B. Note that the purpose of our paper, and of most benchmark papers, is not to establish superior performance of a certain method, but rather to provide an effective and fair ground for comparing different methods.

3) Implementation Details: The proposed model is implemented using PyTorch 1.10.2 [106] and PyTorch Geometric 2.0.3 [107]. A Quadro RTX 8000 GPU with 48GB of memory is used for model training. The optimizer we used is Adam. We train all of our models through 20 epochs, and the learning rate is 1e-3. We use a weight decay of 1e-4 as a means of regularization. The loss function is cross entropy. Hyper-parameters are selected automatically with an open-source AutoML toolkit NNI[19]. Please refer to our repository for comprehensive parameter configurations. When tuning the hyperparameters, we first split the dataset into a train set and a test set with the ratio of 8:2. The k-fold validation is performed on the train set, where we further divide the train set into 10 parts and take one in each run to use as the validation set. The selection of the best hyperparameter is based on the average performance of the model on the validation sets. The reported metrics in Table II, on the other hand, is the average performance on the test set, with each run trained on different train sets. The competing methods are also tuned in the same way. For BrainGNN, we used the author's open-source code[20]. For BrainNetCNN, we implemented it by ourselves with PyTorch, which is publicly available in our BrainGB package[21]. For the hyper-parameter tuning, we selected several important hyper-parameters and performed the grid search on them based on the provided best setting as claimed in their

---

[19] https://github.com/microsoft/nni
[20] https://github.com/xxlya/BrainGNN_Pytorch
[21] https://github.com/HennyJie/BrainGB

paper. To be specific, for BrainGNN, we searched for different learning rates in {0.01, 0.005, 0.001} with different feature dimensions in {100, 200} and the number of GNN layers in {2, 3}. For BrainNetCNN, we searched for different dropout rates in {0.3, 0.5, 0.7} with learning rates in {0.001, 0.0005, 0.0001} and the number of layers in MLP in {1, 2, 3}. The reported results of these two baselines in Table II are from the best performing groups, where for BrainGNN, the learning rate is 0.01, the feature dimension is 200 and the number of GNN layers is 2, and for BrainNetCNN, the dropout rate is 0.3, the learning rate is 0.0001 and the number of layers in MLP is 3. 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.

B. Performance Report

1) Node Feature: On comparing node features, we set the other modules as the well-performed settings in individual tests. Specifically, we use *node edge concat* in Eq. 7 as the message passing scheme, and *concat pooling* in Eq. 18 as the pooling strategy. Our experimental results demonstrate that the **connection profile** which uses the corresponding row in the adjacency matrix as the node features achieves the best performance across all datasets, with up to 33.99% improvements over the second-best, *degree*, on ABCD. We believe this is because the *connection profile* captures the whole picture of structural information in the brain network, and preserves rich information on pairwise connections that can be used to perform brain parcellation. In general, the structure node features (e.g., *degree*, *connection profile*) perform better than the positional ones (e.g., *identity*, *eigen*), indicating that the overall structural information of graph and the structural role of each node are important in the task of brain network analysis. This conclusion is consistent with previous findings in the literature that structural artificial node features work well for graph-level tasks on general graphs [71].

2) Message Passing: To study the effectiveness of different message passing schemes, we initialize the node features with *connection profile* and apply the *concat pooling* to produce graph-level representations, which both perform best when examined separately in each module. Our results reveal that **node concat** (Eq. 8) message passing has the highest AUC performance across four datasets, followed by *node edge concat* (Eq. 7), which achieves a similar AUC performance with sometimes slightly better accuracy and F1 scores (ABCD). The performance superiority of the last two methods may arise from their advantage of reinforcing self-representation of the central node during each step of message passing. This helps to retain the original information from the last step and avoid overfitting towards a biased direction in the optimization process. Surprisingly, the edge involved *node edge concat* performs slightly worse than the pure *node concat*, though the gap gets closer on larger datasets. This indicates that encoding edge weights as a single value may not be useful when the global structure has already been used as the initial node features.

3) Attention Enhanced Message Passing: When evaluating the effectiveness of different attention-enhanced message passing schemes, we set the node features as *connection profile* and apply the *concat pooling* strategy, just as for the evaluation of message passing without attention mechanisms. It is shown that **node concat w/ attn** (Eq. 14) and **node edge concat w/ attn** (Eq. 13) yield very close results across four datasets and they alternately perform the best. Furthermore, the attention-enhanced version achieves better outcomes most of the time (up to 5.23% relative improvements) vs. the corresponding message passing architecture without attention. This demonstrates the effectiveness of utilizing learnable attention weights in the GNN aggregation and update process in addition to the fixed edge weights. Also, *node edge concat w/ attn* surpasses *node concat w/ attn* on larger datasets (e.g., ABCD), which may imply potential advantages of involving edge weights into message design when there are enough training samples.

4) Pooling Strategies: For studying pooling strategies, we employ the *node edge concat* (Eq. 7) as the message passing scheme and *connection profile* as the initial node features. Our findings reveal that the **concat pooling** strategy (Eq. 18) consistently outperforms the other two methods across all four datasets. This is likely because when *concat* is used, the final node representations of all the brain regions are kept in the graph-level representation for classifiers. The other two paradigms, on the other hand, obtain a graph-level embedding with the same dimension of node features. Thus they lose some information that could be helpful for graph-level prediction tasks. Though *concat* does not ensure permutation invariance, it is actually not needed for brain network analysis since the node order given a parcellation is fixed. The compared hierarchical pooling method *DiffPool* demonstrates some advantages on the small HIV dataset but fails to surpass the simple *concat* pooling on three other larger datasets.

5) Other Baselines: In general, we expect deep models like GNNs to perform better on larger datasets. For example, the performance of GNN models on the ABCD dataset clearly surpasses all shallow models by about 2 percent. However, this trend should not prohibit one from experimenting with GNN models on smaller datasets. GNNs do perform well on some small datasets, such as the HIV dataset. Despite running on a small dataset, GNN models in BrainGB have an over 5% advantage over all shallow models. As for the deep baselines, BrainGNN can be out-of-memory (OOM) on large datasets. The best combination based on our modular design outperforms BrainGNN on all four datasets (HIV, PNC, PPMI and ABCD) and achieves comparable results with BrainNetCNN in most cases especially on smaller datasets. These findings prove the need to carefully experiment with our modular designs of GNNs before further developing more complicated architectures, which might overfit certain datasets.

6) Insights on Density Levels: Functional connectivity and structural connectivity have distinctive differences in sparsity levels. Functional networks like ABCD are fully connected. Structural networks like PPMI contain approximately 22.64% edges on average. Through our experiments, we found sparsity levels do have an impact on the choices of hyperparameters. For example, GNNs on the sparser structural networks of PPMI

reach the maximum performance with a hidden dimension of 64, whereas on the functional network of ABCD, they have an optimal hidden dimension of 256, which indicates that GNN models should more complicated with more learnable parameters when the input networks are denser. This observation can be instructive for designing GNN architectures on brain networks constructed from different modalities.

VI. OPEN SOURCE BENCHMARK PLATFORM

To foster future research, we provide an out-of-box package that can be directly installed through pip, with installation and tutorials on our hosted BrainGB website `https://brai ngb.us`. The BrainGB package is also open-sourced at `ht tps://github.com/HennyJie/BrainGB`. We provide examples of GNN-based brain network analysis, trained models, and instructions on imaging preprocessing and functional and structural brain networks construction from raw fMRI and dMRI respectively. It is noted that due to the modular designs, BrainGB can also be extended to other tasks, by adding task-specific functions in each module.

VII. DISCUSSION AND EXTENSIONS

In this paper, we first present BrainGB, a *unified*, *modular*, *scalable*, and *reproducible* framework for brain network analysis with GNNs. While the dataset generation, baselines, and evaluations we provide in BrainGB are thorough, we consider several limitations in the current paradigm:

- The aggregation mechanism in GNN is known to be effective for node-level tasks with the effect of node feature smoothing, and for graph-level tasks due to its capability in structure differentiation. However, for brain networks, what kinds of graph structures (e.g., communities, subgraphs) are effective beyond the pairwise connections are still unknown.
- The small size of neuroimaging datasets may limit the effectiveness and generalization ability of complex deep learning models.

Towards these two limitations, we envision several future directions that can be potentially helpful to fully unleash the power of GNNs for brain network analysis:

- Neurology-driven GNN designs: to design the GNN architectures based on neurological understandings of predictive brain signals, especially disease-specific ones.
- Pre-training and transfer learning of GNNs: to design techniques that can train complex GNN models across studies and cohorts [108]. Besides, information sharing across different diseases could lead to a better understanding of cross-disorder commonalities.

ACKNOWLEDGMENT

REFERENCES

[1] X. Li, Y. Zhou, N. Dvornek, M. Zhang, S. Gao, J. Zhuang, D. Scheinost, L. H. Staib, P. Ventola, and J. S. Duncan, "Braingnn: Interpretable brain graph neural network for fmri analysis," *Med Image Anal*, vol. 74, 2021. 1, 3, 9

[2] F. V. Farahani, W. Karwowski, and N. R. Lighthall, "Application of graph theory for identifying connectivity patterns in human brain networks: a systematic review," *Front. Neurosci.*, vol. 13, p. 585, 2019. 1

[3] K. Osipowicz, M. R. Sperling, A. D. Sharan, and J. I. Tracy, "Functional mri, resting state fmri, and dti for predicting verbal fluency outcome following resective surgery for temporal lobe epilepsy," *J. Neurosurg.*, vol. 124, pp. 929–937, 2016. 1

[4] L. A. Maglanoc, T. Kaufmann, R. Jonassen, E. Hilland, D. Beck, N. I. Landrø, and L. T. Westlye, "Multimodal fusion of structural and functional brain imaging in depression using linked independent component analysis," *Hum Brain Mapp*, vol. 41, pp. 241–255, 2020. 1, 5

[5] E. Bullmore and O. Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems," *Nat. Rev. Neurosci.*, vol. 10, pp. 186–198, 2009. 1, 3

[6] O. Sporns, "Graph theory methods: applications in brain networks," *Dialogues Clin. Neurosci.*, vol. 20, pp. 111–121, 2022. 1

[7] Y. Liu *et al.*, "Multi-view multi-graph embedding for brain network clustering analysis," in *AAAI*, 2018, pp. 117–124. 1, 8

[8] L. Zhan, Y. Liu, Y. Wang, J. Zhou, N. Jahanshad, J. Ye, P. M. Thompson, and A. D. N. I. (ADNI), "Boosting brain connectome classification accuracy in alzheimer's disease using higher-order singular value decomposition," *Frontiers in neuroscience*, vol. 9, p. 257, 2015. 1

[9] J. Faskowitz, R. F. Betzel, and O. Sporns, "Edges in brain networks: Contributions to models of structure and function," *Network Neuroscience*, vol. 6, no. 1, pp. 1–28, 2022. 1, 3

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021. 1

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021, pp. 8748–8763. 1

[12] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *ICCV*, 2021, pp. 6816–6826. 1

[13] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH*, 2020, pp. 5036–5040. 1

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017. 1, 3, 6

[15] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019. 3

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018. 1, 7

[17] J. Kawahara, C. J. Brown, S. P. Miller, B. G. Booth, V. Chau, R. E. Grunau, J. G. Zwicker, and G. Hamarneh, "Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment," *NeuroImage*, vol. 146, pp. 1038–1049, 2017. 1, 3, 9

[18] G. K. Murugesan, C. G. B. Yogananda, S. S. Nalawade, E. M. Davenport, B. C. Wagner, W. H. Kim, and J. A. Maldjian, "Brainnet: Inference of brain network topology using machine learning," *Brain Connect*, vol. 10, pp. 422–435, 2020. 2

[19] C. Su, Z. Xu, J. Pathak, and F. Wang, "Deep learning in mental health outcome research: a scoping review," *Transl. Psychiatry*, vol. 10, no. 1, pp. 1–26, 2020. 2

[20] T. D. Satterthwaite, D. H. Wolf, D. R. Roalf, K. Ruparel, G. Erus, S. Vandekar, E. D. Gennatas, M. A. Elliott, A. Smith, H. Hakonarson *et al.*, "Linked sex differences in cognition and functional connectivity in youth," *Cereb. Cortex*, vol. 25, pp. 2383–2394, 2015.

[21] G. Deco, V. K. Jirsa, and A. R. McIntosh, "Emerging concepts for the dynamical organization of resting-state activity in the brain," *Nat. Rev. Neurosci.*, vol. 12, pp. 43–56, 2011.

[22] Y. Wang and Y. Guo, "A hierarchical independent component analysis model for longitudinal neuroimaging studies," *NeuroImage*, vol. 189, pp. 380–400, 2019.

[23] R. Yu, L. Qiao, M. Chen, S.-W. Lee, X. Fei, and D. Shen, "Weighted graph regularized sparse brain network construction for mci identification," *Pattern Recognit*, vol. 90, pp. 220–231, 2019. 2

[24] T. R. Insel and B. N. Cuthbert, "Brain disorders? precisely," *Science*, vol. 348, pp. 499–500, 2015. 2

[25] L. M. Williams, "Precision psychiatry: a neural circuit taxonomy for depression and anxiety," *Lancet Psychiatry*, vol. 3, pp. 472–480, 2016. 2

[26] W. Li, M. Wang, Y. Li, Y. Huang, and X. Chen, "A novel brain network construction method for exploring age-related functional reorganization," *Comput. Intell. Neurosci.*, vol. 2016, p. 5, 2016. 2

[27] J. Zimmermann, J. D. Griffiths, and A. R. McIntosh, "Unique mapping of structural and functional connectivity on cognition," *J. Neurosci.*, vol. 38, pp. 9658–9667, 2018. 3

[28] Y. Hu, M. Zeydabadinezhad, L. Li, and Y. Guo, "A multimodal multilevel neuroimaging model for investigating brain connectome development," *J Am Stat Assoc*, vol. 117, pp. 1–15, 2022. 3

[29] G. Martensson, J. B. Pereira, P. Mecocci, B. Vellas, M. Tsolaki, I. Kłoszewska, H. Soininen, S. Lovestone, A. Simmons, G. Volpe *et al.*, "Stability of graph theoretical measures in structural brain networks in alzheimer's disease," *Sci. Rep.*, vol. 8, pp. 1–15, 2018. 3

[30] N. Yahata, J. Morimoto, R. Hashimoto, G. Lisi, K. Shibata, Y. Kawakubo, H. Kuwabara, M. Kuroda, T. Yamada, F. Megumi *et al.*, "A small number of abnormal brain connections predicts adult autism spectrum disorder," *Nat. Commun.*, vol. 7, pp. 1–12, 2016.

[31] M. A. Lindquist, "The statistical analysis of fmri data," *Stat Sci*, vol. 23, pp. 439–464, 2008. 3

[32] S. M. Smith, "The future of fmri connectivity," *NeuroImage*, vol. 62, pp. 1257–1266, 2012. 3

[33] R. Shi and Y. Guo, "Investigating differences in brain functional networks using hierarchical covariate-adjusted independent component analysis," *Ann Appl Stat*, p. 1930, 2016.

[34] T. Dai, Y. Guo, A. D. N. Initiative *et al.*, "Predicting individual brain functional connectivity using a bayesian hierarchical model," *NeuroImage*, vol. 147, pp. 772–787, 2017.

[35] I. A. Higgins, S. Kundu, K. S. Choi, H. S. Mayberg, and Y. Guo, "A difference degree test for comparing brain networks," *Hum Brain Mapp*, pp. 4518–4536, 2019. 3

[36] B. Jie, M. Liu, X. Jiang, and D. Zhang, "Sub-network based kernels for brain network classification," in *ICBC*, 2016, pp. 622–629. 3

[37] L. He, K. Chen, W. Xu, J. Zhou, and F. Wang, "Boosted sparse and low-rank tensor regression," in *NeurIPS*, 2018, pp. 1017–1026. 3

[38] Y. Liu, L. He, B. Cao, P. Yu, A. Ragin, and A. Leow, "Multi-view multi-graph embedding for brain network clustering analysis," in *AAAI*, 2018, pp. 117–124. 3, 8

[39] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *ESWC*, 2018, pp. 593–607. 3

[40] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *AAAI*, 2019, pp. 346–353. 3

[41] X. Li, N. C. Dvornek, Y. Zhou, J. Zhuang, P. Ventola, and J. S. Duncan, "Graph neural network for interpreting task-fmri biomarkers," in *MICCAI*, 2019, pp. 485–493. 3

[42] A. Bessadok, M. A. Mahjoub, and I. Rekik, "Graph neural networks in network neuroscience," *TPAMI*, pp. 1–18, 2022.

[43] H. Cui, W. Dai, Y. Zhu, X. Li, L. He, and C. Yang, "Interpretable graph neural networks for connectome-based brain disorder analysis," in *MICCAI*, 2022, pp. 375–385.

[44] Y. Zhu, H. Cui, L. He, L. Sun, and C. Yang, "Joint embedding of structural and functional brain networks with graph neural networks for mental illness diagnosis," in *EMBC*, 2022, pp. 272–276.

[45] X. Kan, H. Cui, L. Joshua, Y. Guo, and C. Yang, "Fbnetgen: Task-aware gnn-based fmri analysis via functional brain network generation," in *MIDL*, 2022.

[46] H. Tang, L. Guo, X. Fu, B. Qu, P. M. Thompson, H. Huang, and L. Zhan, "Hierarchical brain embedding using explainable graph learning," in *ISBI*, 2022, pp. 1–5.

[47] H. Tang, L. Guo, X. Fu, B. Qu, O. Ajilore, Y. Wang, P. M. Thompson, H. Huang, A. D. Leow, and L. Zhan, "A hierarchical graph learning model for brain network regression analysis," *Frontiers in Neuroscience*, vol. 16, pp. 1–5, 2022. 3

[48] S. Sarraf and J. Sun, "Functional brain imaging: A comprehensive survey," *arXiv.org*, 2016. 3

[49] A. Bernstein, R. Akzhigitov, E. Kondrateva, S. Sushchinskaya, I. Samotaeva, and V. Gaskin, "MRI brain imagery processing software in data analysis," *Trans. Mass Data Anal. Images Signals*, vol. 9, pp. 3–17, 2018. 3

[50] G. Ganis and S. M. Kosslyn, "Neuroimaging," in *Encyclopedia of the Human Brain*, 2002, pp. 493–505. 3

[51] E. W. Lang, A. M. Tomé, I. R. Keck, J. M. G. Sáez, and C. G. Puntonet, "Brain connect analysis: A short survey," *Comput. Intell. Neurosci.*, vol. 2012, pp. 412 512:1–412 512:21, 2012. 3

[52] P. Bellec, C. Chu, F. Chouinard-Decorte, Y. Benhajali, D. S. Margulies, and R. C. Craddock, "The neuro bureau adhd-200 preprocessed repository," *NeuroImage*, vol. 144, pp. 275–286, 2017. 4

[53] R. C. Petersen, P. Aisen, L. A. Beckett, M. Donohue, A. Gamst, D. J. Harvey, C. Jack, W. Jagust, L. Shaw, A. Toga *et al.*, "Alzheimer's disease neuroimaging initiative (adni): clinical characterization," *Neurology*, vol. 74, pp. 201–209, 2010. 4, 5

[54] D. C. Van Essen, K. Ugurbil, E. Auerbach, D. Barch, T. E. Behrens, R. Bucholz, A. Chang, L. Chen, M. Corbetta, S. W. Curtiss *et al.*, "The human connectome project: a data acquisition perspective," *NeuroImage*, vol. 62, pp. 2222–2231, 2012. 4, 5

[55] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto *et al.*, "The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism," *Mol. Psychiatry*, vol. 19, pp. 659–667, 2014. 4

[56] Y. Wang, J. Kang, P. B. Kemmer, and Y. Guo, "An efficient and reliable statistical method for estimating functional connectivity in large scale brain networks using partial correlation," *Front. Neurosci.*, vol. 10, p. 123, 2016. 4

[57] S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich, "Network modelling methods for fmri," *NeuroImage*, vol. 54, pp. 875–891, 2011. 4

[58] K. Marek, D. Jennings, S. Lasch, A. Siderowf, C. Tanner, T. Simuni, C. Coffey, K. Kieburtz, E. Flagg, S. Chowdhury *et al.*, "The parkinson progression marker initiative (ppmi)," *Prog. Neurobiol.*, vol. 95, pp. 629–635, 2011. 5

[59] K. A. Ellis, A. I. Bush, D. Darby, D. De Fazio, J. Foster, P. Hudson, N. T. Lautenschlager, N. Lenzo, R. N. Martins, P. Maruff *et al.*, "The australian imaging, biomarkers and lifestyle (aibl) study of aging: methodology and baseline characteristics of 1112 individuals recruited for a longitudinal study of alzheimer's disease," *Int Psychogeriatr*, vol. 21, pp. 672–687, 2009. 5

[60] A. D. Leow, S. Zhu, L. Zhan, K. McMahon, G. I. de Zubicaray, M. Meredith, M. Wright, A. Toga, and P. Thompson, "The tensor distribution function," *Magn Reson Med*, vol. 61, pp. 205–214, 2009. 5

[61] T. E. Behrens, M. W. Woolrich, M. Jenkinson, H. Johansen-Berg, R. G. Nunes, S. Clare, P. M. Matthews, J. M. Brady, and S. M. Smith, "Characterization and propagation of uncertainty in diffusion-weighted mr imaging," *Magn Reson Med*, vol. 50, pp. 1077–1088, 2003. 5

[62] P. J. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi, "In vivo fiber tractography using dt-mri data," *Magn Reson Med*, vol. 44, pp. 625–632, 2000. 5

[63] T. E. Behrens, H. J. Berg, S. Jbabdi, M. F. Rushworth, and M. W.