Computing and Informatics, Vol. 44, 2025, 1040–1070, doi: 10.31577/cai_2025_5_1040

HIBIGNN: HIERARCHICAL BILATERAL GRAPH NEURAL NETWORK FOR FMRI ANALYSIS

Zhengyuan Fan, Hongfei Ji*, Jie Li*

Translational Research Center
Shanghai Yangzhi Rehabilitation Hospital
(Shanghai Sunshine Rehabilitation Center)
School of Computer Science and Technology
Tongji University, Shanghai 201804, China
e-mail: {fzy, jhf}@tongji.edu.cn, nijanice@163.com

Jie Zhuang

School of Psychology
Shanghai University of Sport, Shanghai 200438, China
e-mail: jie.zhuang@sus.edu.cn

Qian Qian*

Department of Neurology and Neurological Rehabilitation
Shanghai Yangzhi Rehabilitation Hospital
(Shanghai Sunshine Rehabilitation Center)
School of Medicine, Tongji University, Shanghai 201619, China
e-mail: flowersinmyheart@126.com

Abstract. Graph Neural Networks (GNNs) have shown great promise in functional Magnetic Resonance Imaging (fMRI) analysis due to their ability to capture complex interactions between brain regions. However, existing models often overlook the brain's physiological structure and fail to leverage hierarchical information from brain atlases. In this paper, we propose Hierarchical Bilateral Graph Neural Net-

* Corresponding author

work (HiBiGNN), a generic architecture that integrates hierarchical information from brain atlases and incorporates the bilateral structure of the brain, with the ability to be instantiated with various existing GNNs as its foundation. HiBiGNN processes a special heterogeneous graph structure, called the Hierarchical Bilateral Graph (HiBiG), which combines multi-level brain graphs derived from functional regions defined by multi-level brain atlases and divides each brain graph into left and right subgraphs, thereby modeling multiple types of nodes and relations. During feature extraction, HiBiGNN performs deep fusion of features from different types of nodes using a unique convolution operation (HiBiG-Conv) and generates graph-level representations via a specialized readout operation (HiBiG-Readout) for graph classification tasks. To assess the effectiveness of HiBiGNN, we conducted extensive experiments on a graph classification task using an fMRI dataset we collected from a response inhibition task, testing multiple HiBiGNN instances with different base GNN models. The results show that our HiBiGNN instances outperforms several generic GNN models as well as those specifically designed for fMRI analysis, demonstrating the significant potential of HiBiGNN for future applications.

Keywords: fMRI, GNN, heterogeneous graph, graph classification, brain network

Mathematics Subject Classification 2010: 68-T10

1 INTRODUCTION

Unraveling the functional mechanisms of the brain is a key objective of modern neuroscience. To achieve this goal, researchers utilize advanced imaging techniques and computational models to capture and interpret complex brain activity patterns. Among these techniques, functional Magnetic Resonance Imaging (fMRI) can record brain activity with high spatial resolution, making it widely used in cognitive neuroscience, psychology, clinical medicine, and other fields.

Using fMRI data to construct functional brain networks is a common analysis approach, where the brain networks are modeled as graphs, with nodes corresponding to brain regions and edges representing their connectivity. Typically, the brain regions are referred to as Regions of Interest (ROIs). Traditional brain network analysis involves manually defining graph features based on graph-theoretical metrics. However, the manual extraction of features demands extensive expertise, and the quality of the defined features can greatly impact the effectiveness of subsequent analyses.

In recent years, Graph Neural Networks (GNNs) have gained increasing popularity for their exceptional ability to handle graph-structured data. Representative works include GCN [1], GraphSAGE [2], GAT [3], GIN [4], and others. GNNs can automatically learn high-level feature representations for nodes, edges, and graphs from raw input data, avoiding the need for complex feature engineering and capturing intricate patterns more effectively. As a result, GNNs have achieved excellent

performance across a wide range of applications, including social recommendation, knowledge graph completion, drug discovery, traffic prediction, text classification, and others.

The remarkable performance of GNNs has made them a popular choice for fMRI analysis, with various models being proposed and applied to different tasks, particularly graph classification tasks, such as gender prediction, cognitive states classification, and classification between neurological disorder patients and healthy controls.

Using GNNs for fMRI-based graph classification generally involves two steps: constructing graphs from fMRI data, and applying GNNs to extract graph-level representations. Typically, a single brain atlas is used in the first step to parcellate brain regions. To construct a brain graph, the average fMRI time series for each region are extracted to derive node features, and the functional connectivity (FC) between regions is calculated to define the edges. Since the nodes in the graph correspond to brain regions from a single atlas, this graph structure is flat and does not have a hierarchical organization.

However, from a neuroscience perspective, the brain can be parcellated at different scales, forming a tree-like hierarchical structure. At a coarser level, the brain can be divided into major lobes, such as the frontal lobe, parietal lobe, and temporal lobe. At a finer level, a lobe can be subdivided into gyrus-level regions: for example, the frontal lobe comprises regions such as the inferior frontal gyrus, middle frontal gyrus, and superior frontal gyrus. Moreover, a gyrus-level region can be further subdivided into subregions: for instance, the inferior frontal gyrus can be partitioned into the orbital part, triangular part, and opercular part.

Brain networks built on parcellations at varying scales incorporate information at different levels. Inspired by the hierarchical structure of brain parcellations in neuroscience, we propose that constructing and integrating multi-level brain networks, rather than relying on a single level, can yield richer graph-level feature representations.

Additionally, existing research typically treats the brain as a unified entity when constructing brain networks, assuming uniform information exchange mechanisms across all brain regions, without distinguishing between the left and right hemispheres. However, from a structural standpoint, the two cerebral hemispheres have a degree of independence and are interconnected through the corpus callosum. Neuroscientific studies indicate that the left and right hemispheres are functionally asymmetric, with the corpus callosum facilitating communication between them to ensure smooth overall functioning.

Considering the physiological structure and collaborative pattern of the left and right hemispheres, we propose that when constructing brain networks, treating the internal connections within each hemisphere and the connections between them as distinct relations and applying graph convolution operations with different parameters can better simulate the pattern of information exchange between left and right brain regions, thereby leading to more meaningful node-level feature representations.

By simultaneously considering the hierarchical parcellations of the brain and the interaction pattern between the left and right hemispheres, we propose that

the brain network can be represented as a Hierarchical Bilateral Graph (HiBiG), which contains multiple types of nodes and multiple types of connections. Correspondingly, we have developed the Hierarchical Bilateral Graph Neural Network (HiBiGNN) that takes HiBiG as input, fusing features from different types of nodes and relations during the graph learning process to obtain more comprehensive feature representations.

The main contributions of this paper are as follows:

1. We introduce the HiBiG structure for fMRI-based brain network construction, addressing the underutilization of hierarchical information from brain atlases and the neglect of the brain's physiological structure.
2. We propose the HiBiGNN architecture, which takes HiBiG as input and can be instantiated with existing GNNs, integrating multi-level information and simulating interaction between left and right brain hemispheres.
3. The proposed method has been validated on a self-collected fMRI dataset and outperforms several existing models, demonstrating the effectiveness of HiBiG structure and HiBiGNN architecture.

2 RELATED WORK

2.1 Graph Neural Networks for fMRI Analysis

In recent years, GNNs have been increasingly used for fMRI-based brain network analysis. Numerous studies have concentrated on deriving graph-level representations from static brain networks and identifying important brain regions for specific tasks. Yan et al. [5] proposed GroupINN, which introduces a node grouping layer to coarsen brain networks, significantly reducing the number of model parameters and identifying the most task-relevant brain subnetworks in prediction tasks. Li et al. [6] introduced BrainGNN, which considers the variability between ROIs by designing an ROI-aware graph convolutional layer that applies different kernel weights to different ROIs, and uses an ROI-selection pooling layer to identify important ROIs. Cui et al. [7] proposed IBGNN, which uses an edge-weight-aware message passing mechanism to address the issue that both positive and negative values may occur in correlation-based edge weights, and introduces an interpretable module that learns a globally shared edge mask to identify important brain regions and connections. Kan et al. [8] introduced FBNetGen, which incorporates a learnable brain graph generation module that integrates the generation of brain graphs and the training of GNNs into an end-to-end pipeline, making the generated brain graphs more relevant to specific downstream tasks compared to the traditional two-stage method where brain graphs are first manually constructed and then GNNs are trained. Furthermore, Kan et al. [9] presented Brain Network Transformer, which combines the characteristics of GNNs and Transformers by using attention mechanisms to learn the connection strengths between ROIs and considering modular-level similarities

within brain networks, and introduces an orthonormal clustering readout operation that generates cluster-aware node embeddings through unsupervised soft clustering and orthonormal projection, leading to more meaningful graph-level representations.

Additionally, some studies focus on extracting both temporal and spatial features simultaneously from dynamic brain graphs. Kim et al. [10] developed STAGIN, which uses dynamic FCs to construct a sequence of brain graphs instead of a static graph as its input, and employs a spatio-temporal attention mechanism to obtain dynamic graph feature representations, along with a novel readout module and Transformer encoder to achieve interpretability in both temporal and spatial dimensions. Yan et al. [11] proposed Multi-Head GAGNN, which uses a spatial multi-head attention graph U-Net to capture the spatial features of multiple brain networks, and introduces a temporal multi-head guided attention network to model the temporal characteristics guided by the extracted spatial information. Liu et al. [12] proposed BrainTGL, which integrates the advantages of GCNs and LSTMs to capture spatio-temporal features from both fMRI time series and dynamic graph structures, and employs an attention-based temporal graph pooling method to eliminate irrelevant functional connections and data inconsistency.

The aforementioned studies have made various explorations in the feature extraction of brain networks. However, they typically use a single brain atlas to divide brain regions and extract fMRI time series, without considering the hierarchical structure of brain functions or the distinctive connectivity patterns between the left and right hemispheres. Given that fMRI data is inherently 4D images, it can be processed into different graph structures, and we believe that designing more informative graph structures can better exploit the potential of fMRI data, which requires the development of backbone models specifically tailored to the new structures for effective feature extraction.

2.2 Heterogeneous Graph Neural Networks

Traditional GNNs operate on homogeneous graphs where all nodes and edges belong to the same type, whereas many real-world applications involve heterogeneous graphs with multiple node and edge types. Heterogeneous Graph Neural Networks (HGNNs) are designed to handle such scenarios by incorporating different types of information and relations within heterogeneous graphs.

To address the complexities of heterogeneous graphs, various efforts in HGNNs focus on extending traditional homogeneous GNNs to accommodate multiple node and edge types. Schlichtkrull et al. [13] proposed R-GCN, which extends the message passing framework of GCN to multi-relational graphs by decomposing the heterogeneous graph into multiple homogeneous subgraphs and learns different transformation matrices for different relations. Zhang et al. [14] developed HetGNN, which handles structural and content heterogeneity in heterogeneous graphs by using different encoders for different content types (e.g., text or images), aggregating neighbor features separately, and combining features using attention mechanisms. Hu et al. [15] introduced HGT, which uses a heterogeneous mutual attention mech-

anism to obtain contextualized node representations, incorporates relative temporal encoding for dynamic graphs, and proposes an efficient mini-batch sampling algorithm for Web-scale graphs.

Additionally, metapath-based approaches constitute a significant branch within the field of HGNN research. Wang et al. [16] proposed HAN, which accounts for the rich semantic information in heterogeneous graphs by designing metapaths and employing hierarchical attention mechanisms at both node-level and metapath-level to learn the importance of nodes and metapaths simultaneously. Fu et al. [17] developed MAGNN, which converts node information into a unified feature space, aggregates structural and semantic information within each metapath, and integrates information from all metapaths using attention mechanisms. Yun et al. [18] introduced GTN, which performs soft selection of edge types to handle noisy graphs, automatically generates metapaths based on the data, and transforms the input graph into useful metapath graphs for convolutions, offering greater scalability than manually defined metapaths.

Currently, there is a lack of application of heterogeneous graphs in brain network analysis. Inspired by R-GCN and other studies, we model brain networks as multi-relational heterogeneous graphs in our study, employing graph convolutional operations with distinct parameters for each relation during message passing, and aggregating messages from various relations to update node features of brain regions.

3 METHODOLOGY

3.1 Hierarchical Bilateral Graph Structure

In this section, we propose the Hierarchical Bilateral Graph (HiBiG) structure, illustrated in Figure 1, which is specifically designed for fMRI-based brain network construction.

The construction of HiBiG involves two steps: the construction of Hierarchical Graph (HiG) and the construction of Bilateral Graph (BiG). The following parts will sequentially introduce HiG and BiG, and then explain how to combine them to obtain HiBiG.

3.1.1 Construction of HiG

HiG is composed of multiple levels of subgraphs, where each subgraph is generated based on different hierarchical levels of brain parcellations, such as lobe-level, gyrus-level, and subregion-level. Within a subgraph, nodes represent brain regions at the corresponding level of parcellation, and the features of these nodes are summaries of fMRI time series for those regions. Connections within a subgraph represent the relation between brain regions at the same level (e.g., between a gyrus and another gyrus), similar to those in single-level graphs of previous studies. Additionally, there are connections between different levels of subgraphs (e.g., between a lobe and a gyrus), representing the relation between regions at different levels of parcellations.

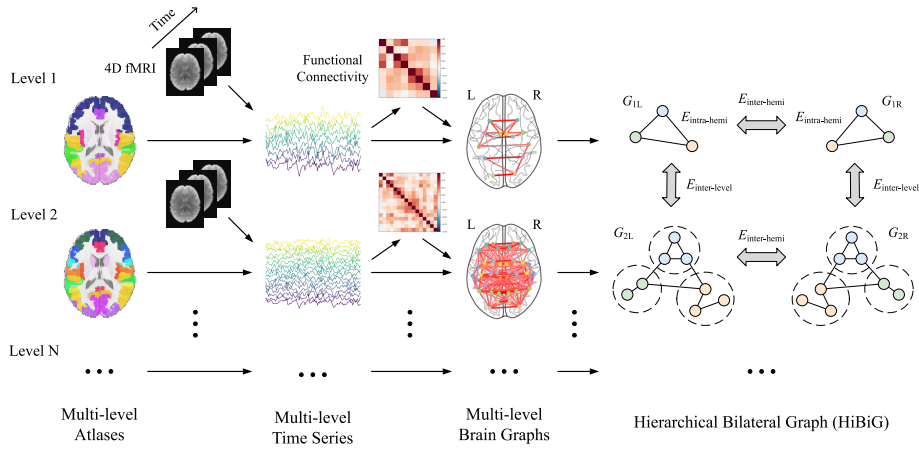


Figure 1. Construction of Hierarchical Bilateral Graph (HiBiG). The construction of the HiBiG begins by extracting multi-level time series from 4D fMRI using multi-level atlases. Functional connectivity matrices are then calculated from time series to establish intra-hemispheric connections. Multi-level brain graphs are then constructed, with each level divided into left and right hemisphere subgraphs, adding inter-hemispheric connections and establishing inter-level connections between subgraphs at different hierarchical levels, resulting in HiBiG.

Formally, we define HiG as a multi-relational graph $G_{\mathrm{H}} = (V_{\mathrm{H}}, E_{\mathrm{H}}, R_{\mathrm{H}})$, where V_{H} is the node set, E_{H} is the edge set, and R_{H} is the relation set. The edge set E_{H} contains both intra-level edges and inter-level edges, and the relation set R_{H} contains the corresponding relations:

$$E_{\mathrm{H}} = E_{\text{intra-level}} \cup E_{\text{inter-level}}, \tag{1}$$

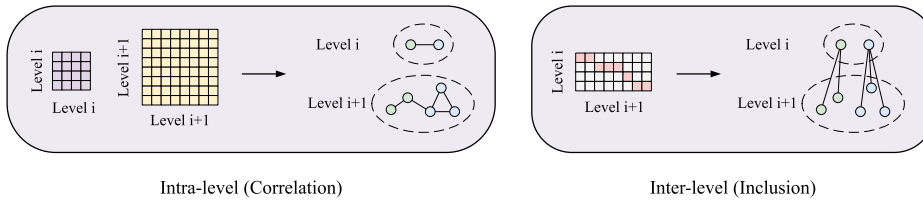
$$R_{\mathrm{H}} = R_{\text{intra-level}} \cup R_{\text{inter-level}}. \tag{2}$$

Specifically, there are L hierarchical levels, and the subgraph at level i is denoted as $G_i = (V_i, E_i)$, which contains only intra-level relation. Consequently, the entire node set V can be expressed as the union of node sets at all levels:

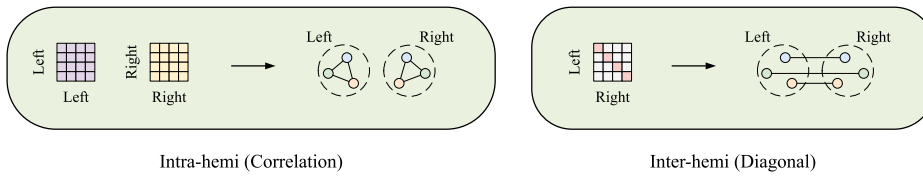
$$V_{\mathrm{H}} = \bigcup_{i=1}^{L} V_i, \tag{3}$$

and the intra-level edge set $E_{\text{intra-level}}$ is the union of edge sets at all levels:

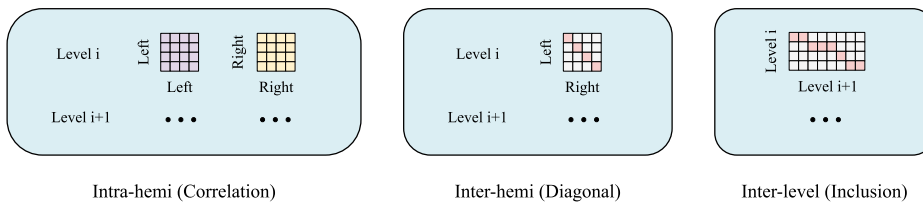
$$E_{\text{intra-level}} = \bigcup_{i=1}^{L} E_i. \tag{4}$$



a) HiG: Including intra-level connections derived from the correlation matrix, and inter-level connections defined by the inclusion relation



b) BiG: Including intra-hemispheric connections derived from the correlation matrix, and inter-hemispheric connections defined by a diagonal matrix



c) HiBiG: Combination of HiG and BiG, including intra-hemispheric connections, inter-hemispheric connections, and inter-level connections

Figure 2. Connections in HiG, BiG and HiBiG

Given the hierarchical structure of HiG, constructing it requires multi-level fMRI time series as input, which can be denoted as $X = [X_1, X_2, \ldots, X_{\mathrm{L}}]$, where $X_i \in \mathbb{R}^{N_i \times d}$. Here, N_i is the number of regions at level i, and d is the length of the time series, which corresponds to the input feature size for each node.

To establish intra-level connections at level i, the correlation matrix $C_i \in \mathbb{R}^{N_i \times N_i}$ is computed from the time series of all regions at this level. The intra-level adjacency matrix $A_i \in [0,1]^{N_i \times N_i}$ is derived from C_i by applying a threshold to retain significant connections. The computation of C_i and A_i is as follows:

$$C_i = \mathrm{Correlation}(X_i), \tag{5}$$

$$A_i(l,k) = \begin{cases} 1, & \text{if } C_i(l,k) > \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Here, Correlation$(\cdot)$ is used to compute the correlation matrix of X_i, with available metrics including Pearson correlation, partial correlation, etc., and $C_i(l,k)$ represents the correlation value between regions l and k at level i. Edges in C_i are filtered by (6) based on a specified threshold τ, such that only connections with correlation values greater than τ are retained in the adjacency matrix A_i. It is important to note that the correlation matrix C_i can contain negative values, representing anti-correlated regions. However, in this study, the threshold τ is set to 0, and therefore, negative correlations are not considered.

The connections between levels (i.e., inter-level connections) are defined by the inclusion relation of regions, as shown in Figure 2 a). For example, the inferior frontal gyrus (IFG) is part of the frontal lobe, thus a connection is established between the IFG and the frontal lobe. Assuming i represents a coarser level and j represents a finer level, the inter-level adjacency matrix $A_{ij} \in [0,1]^{N_i \times N_j}$ between level i and level j is defined as follows:

$$A_{ij}(l,k) = \begin{cases} 1, & \text{if } B_{i,l} \supseteq B_{j,k}, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $B_{i,l}$ and $B_{j,k}$ represent brain regions l and k at levels i and j, respectively.

In short, HiG is a multi-level graph structure that features two types of connections: intra-level and inter-level connections. This design allows HiG to not only handle interactions between brain regions at the same level, but also capture the hierarchical relation between regions at different levels.

3.1.2 Construction of BiG

BiG divides a single-level graph structure into two subgraphs, representing the left and right hemispheres, respectively. Nodes are assigned to the corresponding subgraph based on whether they belong to the left or right hemisphere. Connections within each subgraph constitute intra-hemispheric connections, whereas connections between the two subgraphs constitute inter-hemispheric connections.

Formally, we define BiG as a multi-relational graph $G_{\mathrm{B}} = (V_{\mathrm{B}}, E_{\mathrm{B}}, R_{\mathrm{B}})$, which can be derived from a traditional single-level brain graph $G_{\mathrm{trad}} = (V_{\mathrm{trad}}, E_{\mathrm{trad}})$ by first partitioning it into left and right hemispheric subgraphs $G_{\mathrm{L}} = (V_{\mathrm{L}}, E_{\mathrm{L}})$ and $G_{\mathrm{R}} = (V_{\mathrm{R}}, E_{\mathrm{R}})$, and then handling intra- and inter-hemispheric connections separately.

Nodes in V_{trad} are assigned to either V_{L} or V_{R} based on their anatomical locations. For simplicity, we assume that each node in V_{L} has a corresponding node in V_{R}, thus $N_{\mathrm{L}} = N_{\mathrm{R}} = \frac{N}{2}$, where N is the total number of nodes in G_{trad}. Consequently, the node set V_{B} of BiG is defined as the union of node sets from both hemispheres, and it is identical to V_{trad}:

$$V_{\mathrm{B}} = V_{\mathrm{trad}} = V_{\mathrm{L}} \cup V_{\mathrm{R}}. \tag{8}$$

The edge set E_{B} contains both intra-hemispheric edges and inter-hemispheric edges,

and the relation set R_{B} contains the corresponding relations:

$$E_{\mathrm{B}} = E_{\text{intra-hemi}} \cup E_{\text{inter-hemi}}, \tag{9}$$

$$R_{\mathrm{B}} = R_{\text{intra-hemi}} \cup R_{\text{inter-hemi}}. \tag{10}$$

Here, intra-hemispheric edge set $E_{\text{intra-hemi}} = E_{\mathrm{L}} \cup E_{\mathrm{R}}$.

Let $k \in \{\mathrm{L}, \mathrm{R}\}$ denotes the left or right hemisphere, and the intra-hemispheric adjacency matrix $A_k \in [0,1]^{N_k \times N_k}$ represents the connectivity within each hemisphere. Specifically, for each hemisphere, A_k is directly derived from the original adjacency matrix $A_{\text{trad}} \in [0,1]^{N \times N}$ of G_{trad} by selecting the rows and columns corresponding to nodes in V_k:

$$A_k = A[V_k, V_k], \tag{11}$$

where $A[V_k, V_k]$ denotes the submatrix of A with rows and columns indexed by the nodes in V_k. This ensures that the intra-hemispheric connectivity within each subgraph remains consistent with the original graph.

The inter-hemispheric adjacency matrix A_{LR} represents connectivity between left and right hemispheres, as shown in Figure 2 b). When $N_{\mathrm{L}} = N_{\mathrm{R}} = N/2$, we can sort the nodes in V_{L} and V_{R} such that corresponding nodes in the left and right hemispheres have the same index. With this setup, we define the inter-hemispheric adjacency matrix A_{LR} as a diagonal matrix:

$$A_{\mathrm{LR}}(i,j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where only the corresponding connections between left and right brain regions are set to 1, whereas all other connections are set to 0.

By dividing the graph into left and right subgraphs and applying appropriate adjacency matrices, BiG provides a clear framework for modeling both intra- and inter-hemispheric interactions.

3.1.3 Combining HiG and BiG to Form HiBiG

As discussed above, HiG is a hierarchical composition of the traditional single-level graph structure in previous studies, whereas BiG is a division of the single-level graph into two subgraphs corresponding to the left and right hemispheres. These two approaches provide complementary perspectives on brain connectivity: HiG captures the hierarchical organization of brain regions, whereas BiG focuses on the interaction between the left and right hemispheres.

Building on these concepts, HiBiG is a combination of the HiG and BiG structures, consisting of multi-level hierarchical subgraphs, with each hierarchical subgraph further divided into left and right hemispheric subgraphs. This design allows HiBiG to consider both the hierarchical representation of the brain and the interaction pattern between the left and right hemispheres simultaneously.

Formally, we define HiBiG as a multi-relational graph $G_{\mathrm{HB}} = (V_{\mathrm{HB}}, E_{\mathrm{HB}}, R_{\mathrm{HB}})$, where V_{HB} is the node set, E_{HB} is the edge set, and R_{HB} is the relation set. The hierarchical subgraphs are denoted as $G_i = (V_i, E_i, R_i)$, where i indexes different hierarchical levels. Each hierarchical subgraph G_i is divided into two hemispheric subgraphs $G_{i\mathrm{L}} = (V_{i\mathrm{L}}, E_{i\mathrm{L}}, R_{i\mathrm{L}})$ and $G_{i\mathrm{R}} = (V_{i\mathrm{R}}, E_{i\mathrm{R}}, R_{i\mathrm{R}})$, representing the left and right hemispheres.

There are three types of connections in HiBiG, as shown in Figure 2 c), each treated as a distinct type of relation:

Inter-level connections: These connections exist between the hierarchical subgraphs, capturing the hierarchical associations between different levels of brain parcellations. For example, a connection might link a lobe-level brain region to a gyrus-level region within that lobe.

Inter-hemispheric connections: These connections exist between the left and right hemispheric subgraphs at the same level, modeling the interactions between the two hemispheres. For example, a connection might link a brain region in the left hemisphere to the corresponding region in the right hemisphere.

Intra-hemispheric connections: These connections exist within each hemispheric subgraph, representing the internal connectivity within each hemisphere. For example, a connection might link two brain regions within the same hemisphere that have a high correlation.

We denote the three types of connections as $E_{\text{inter-level}}$, $E_{\text{inter-hemi}}$, and $E_{\text{intra-hemi}}$, with the corresponding relation types being $r_{\text{inter-level}}$, $r_{\text{inter-hemi}}$, and $r_{\text{intra-hemi}}$. Therefore, E and R in HiBiG can be expressed as:

$$E = E_{\text{inter-level}} \cup E_{\text{inter-hemi}} \cup E_{\text{intra-hemi}}, \tag{13}$$

$$R = R_{\text{inter-level}} \cup R_{\text{inter-hemi}} \cup R_{\text{intra-hemi}}. \tag{14}$$

Furthermore, the three types of connections can be defined by the following formulas, where $\mathrm{Level}(\cdot)$ and $\mathrm{Hemi}(\cdot)$ represent the hierarchical level and hemisphere of a node:

$$\begin{aligned} E_{\text{inter-level}} = \{(v_i, r, v_j) \mid \mathrm{Level}(v_i) \neq \mathrm{Level}(v_j) \\ \wedge \mathrm{Hemi}(v_i) = \mathrm{Hemi}(v_j), r \in R_{\text{inter-level}}\}, \end{aligned} \tag{15}$$

$$\begin{aligned} E_{\text{inter-hemi}} = \{(v_i, r, v_j) \mid \mathrm{Hemi}(v_i) \neq \mathrm{Hemi}(v_j) \\ \wedge \mathrm{Level}(v_i) = \mathrm{Level}(v_j), r \in R_{\text{inter-hemi}}\}, \end{aligned} \tag{16}$$

$$\begin{aligned} E_{\text{intra-hemi}} = \{(v_i, r, v_j) \mid \mathrm{Hemi}(v_i) = \mathrm{Hemi}(v_j) \\ \wedge \mathrm{Level}(v_i) = \mathrm{Level}(v_j), r \in R_{\text{intra-hemi}}\}. \end{aligned} \tag{17}$$

In short, HiBiG is a heterogeneous graph structure that contains multiple types of nodes and relations. This allows the HiBiGNN architecture to effectively integrate

multi-level information and facilitate interaction between the left and right brain hemispheres.

3.2 Hierarchical Bilateral Graph Neural Network

In this section, we propose the Hierarchical Bilateral Graph Neural Network (HiBiGNN) architecture, illustrated in Figure 3, which is developed to exploit the hierarchical and bilateral structure of HiBiG.

To give an overview, HiBiGNN takes the HiBiGs constructed from fMRI data as input, updates node features through graph convolution, and summarizes graph features for graph classification. Two types of layers play crucial roles in this process: the convolutional layer, denoted as HiBiG-Conv, and the readout layer, denoted as HiBiG-Readout. The HiBiG-Conv layer facilitates information exchange between nodes and their neighbors across multiple relations, and the HiBiG-Readout layer aggregates node features from hemispheric and hierarchical subgraphs to obtain graph-level representations.

Collectively, these components are tailored to take advantage of HiBiG's unique hierarchical and bilateral structure, improving the model's ability to extract comprehensive features from complex fMRI data. The following parts will provide a detail to the components of HiBiGNN.

3.2.1 HiBiG-Conv Layer

As shown in Figure 3, the HiBiG-Conv layer integrates the HiG convolutional layer (HiG-Conv) and the BiG convolutional layer (BiG-Conv), thereby leveraging both the hierarchical and bilateral structure of HiBiG. The HiG-Conv layer can facilitate information exchange across different levels, and the BiG-Conv layer can manage interaction within and between left and right hemispheres. Consequently, the HiBiG-Conv layer can simultaneously handle all three types of relations: intra-hemisphere, inter-hemisphere, and inter-level.

The computations performed by HiBiG-Conv, HiG-Conv, and BiG-Conv layers can be described using unified formulas, where only the relation set R changes, whereas the message passing process remains the same. Specifically, for HiBiG-Conv, $R = R_{\text{intra-hemi}} \cup R_{\text{inter-hemi}} \cup R_{\text{inter-level}}$. The message passing process can be divided into two steps:

Step 1: Generating messages under each relation.

Firstly, for each relation $r \in R$, we use a specific graph convolution operation GConv_r to generate the node message under this relation:

$$a_v^{(m+1)}(r) = \text{GConv}_r\left(h_v^{(m)}, \{h_u^{(m)} \mid u \in \mathcal{N}_r(v)\}\right), \tag{18}$$

where GConv_r is an instance of graph convolution operations from existing GNNs (e.g., GCN, GIN, GAT, GraphSAGE), $h_v^{(m)}$ is the feature of node v at layer m, and $\mathcal{N}_r(v)$ is the neighborhood of node v under relation r.

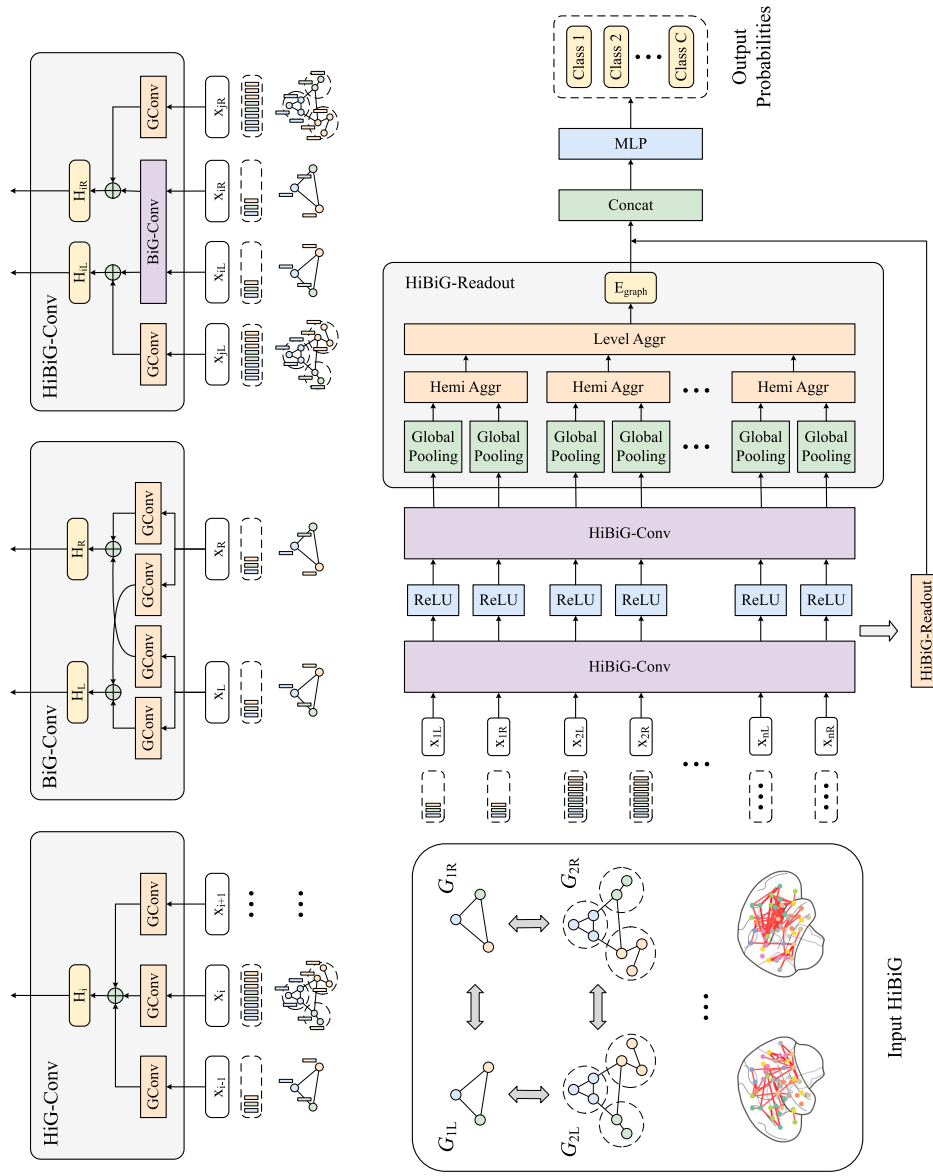


Figure 3. Framework of Hierarchical Bilateral Graph Neural Network (HiBiGNN). HiG-Conv: Fusion of features from multiple hierarchical levels. BiG-Conv: Fusion of features from the left and right hemispheres. HiBiG-Conv: Combination of HiG-Conv and BiG-Conv, fusing features from multiple hierarchical levels and both hemispheres. HiBiGNN: Two stacked HiBiG-Conv layers, each followed by a HiBiG-Readout layer, with the extracted graph feature passed through an MLP to produce the graph classification output

Step 2: Aggregating messages from all relations.

Secondly, we aggregate messages from multiple relations to update node features:

$$h_v^{(m+1)} = \sigma\left(\sum_{r\in R} a_v^{(m+1)}(r)\right), \tag{19}$$

where σ is an activation function (e.g., ReLU).

Combining the above two steps, the complete formula for the multi-relational graph convolution operation (HiBiG-Conv, HiG-Conv, or BiG-Conv) is:

$$h_v^{(m+1)} = \sigma\left(\sum_{r\in R} \mathrm{GConv}_r\left(h_v^{(m)}, \{h_u^{(m)} \mid u \in \mathcal{N}_r(v)\}\right)\right). \tag{20}$$

When drawing Figure 3, we illustrated HiBiG-Conv as a nested structure of HiG-Conv and BiG-Conv to make the diagram clearer, which first fuses intra-hemi and inter-hemi features, and then fuses inter-level features. Since the fusion method is addition, this nested fusion process is equivalent to summing the features from several relations simultaneously, which is consistent with Equation (20).

From the above description, it can be seen that HiBiG-Conv (as well as HiG-Conv and BiG-Conv) allows for various existing graph convolution operations to be used as GConv_r. This flexibility makes HiBiG-Conv a versatile convolution method, adaptable to different graph datasets without being confined to a specific computation process. Specifically, GConv_r for each relation $r \in R$ can either be identical or distinct across all relations depending on the specific requirements of the dataset.

Furthermore, we can easily adapt existing GNN models by replacing their graph convolutional layers with HiBiG-Conv layers, thereby enabling them to handle the HiBiG structure as input. To achieve this, one only needs to apply the same convolution operation across all relations in Equation (20). For instance, to replace the graph convolutional layers in GraphSAGE with HiBiG-Conv, one could set $\mathrm{GConv}_r = \text{SAGE-Conv}_r$.

When constructing a HiBiGNN, a simple approach is to first build an ordinary GNN, referred to as the base GNN, which accepts traditional single-level graphs without hierarchical composition or bilateral division, and then transform it into HiBiGNN by modifying its convolutional layers and readout layers. For example, if a HiBiGNN instance is constructed based on GraphSAGE, we denote it as HiBiGNN-SAGE.

In this study, we first construct several base GNNs and then transform them into HiBiGNNs, where all relations use the same type of GConv_r operation, rather than selecting different GConv_r operations for different relations.

The choice of GConv_r is crucial for HiBiG-Conv, and different datasets may be better suited to different base GNNs. Moreover, combining multiple types of GConv_r might yield better performance.

3.2.2 HiBiG-Readout Layer

The HiBiG-Readout layer is applied after each HiBiG-Conv layers to aggregate the node features of the HiBiG and obtain a graph-level feature representation. For each HiBiG-Readout layer, the readout process involves three steps:

Step 1: Global pooling of hemispheric subgraphs.

Firstly, to summarize the feature of each hemispheric subgraph after the m-th convolutional layer, we adopt two types of global pooling methods: mean pooling and max pooling. The two pooling results are then concatenated to form the feature of the hemispheric subgraph:

$$g_{ik}^{(m)} = \text{mean}\left(H_{ik}^{(m)}\right) \parallel \max\left(H_{ik}^{(m)}\right), \tag{21}$$

where $g_{ik}^{(m)}$ denotes the feature of the hemispheric subgraph G_{ik} obtained from the m-th readout layer, $i \in \{1, \ldots, N_i\}$ denotes the hierarchical level, and $k \in \{L, R\}$ denotes the hemisphere (left or right). $H_{ik}^{(m)} = \left\{h_j^{(m)} \mid v_j \in V_{ik}\right\}$ is the node feature set of node set V_{ik} after the m^{th} convolutional layer. The mean pooling function mean$(\cdot)$ computes the channel-wise average of the node features, and the max pooling function max$(\cdot)$ computes the channel-wise maximum. The operator $\parallel$ denotes the concatenation operation.

Step 2: Aggregation of subgraph features from the left and right hemispheres.

Considering the different contributions of the left and right hemispheres to graph classification tasks, we employ a weighted average approach to fuse the features of the left and right hemispheric subgraphs at each hierarchical level, thereby obtaining the feature for each hierarchical subgraph:

$$g_i^{(m)} = \frac{1}{w_{\text{L}} + w_{\text{R}}}\left(w_{\text{L}} g_{i\text{L}}^{(m)} + w_{\text{R}} g_{i\text{R}}^{(m)}\right), \tag{22}$$

where $g_i^{(m)}$ denotes the feature of the hierarchical subgraph G_i obtained from the m^{th} readout layer, and $w_{\text{L}}, w_{\text{R}}$ are learnable weights representing the importance of the left and right hemispheres. The normalization factor $1/(w_{\text{L}} + w_{\text{R}})$ ensures that the scale of the final feature vector remains consistent. The same set of w_{L} and w_{R} is shared across all hierarchical levels to ensure a consistent contribution from the left and right hemispheres to all hierarchical subgraphs.

Step 3: Aggregation of subgraph features from multiple hierarchical levels.

After obtaining the features of hierarchical subgraphs, we concatenate them to obtain the feature of the overall HiBiG:

$$g^{(m)} = g_1^{(m)} \parallel g_2^{(m)} \parallel \cdots \parallel g_L^{(m)}, \tag{23}$$

where $g^{(m)}$ is the feature of the HiBiG obtained from the m^{th} readout layer, and L is the number of hierarchical levels.

Once the above three steps are completed, we concatenate the graph features obtained from all readout layers to get the feature for graph classification:

$$g_{\text{final}} = g^{(1)} \parallel g^{(2)} \parallel \cdots \parallel g^{(M)}, \tag{24}$$

where M is the number of readout layers.

Finally, we input g_{final} into a Multilayer Perceptron (MLP) to obtain the classification result:

$$z = \text{MLP}(g_{\text{final}}), \tag{25}$$

where $z \in \mathbb{R}^C$ represents the probability distribution over the C classes, and the class with the highest probability is selected as the predicted class.

3.2.3 Loss Function

In this study, we simply use the cross-entropy loss for training, which is a common choice for classification tasks. The cross-entropy loss measures the discrepancy between the predicted probability distributions and the true labels. For a given sample with true label y and predicted probability distribution z, the cross-entropy loss is defined as:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(z_i), \tag{26}$$

where y_i is the binary indicator (0 or 1) for class i, z_i is the predicted probability for class i, and C is the number of classes.

Minimizing this loss helps the model adjust parameters, boost correct class probabilities, and penalize errors, improving classification performance.

4 EXPERIMENTS AND RESULTS

4.1 Dataset

We used a self-collected event-related fMRI dataset from a response inhibition task, specifically a Go/No-Go lexical decision task, to assess the effectiveness of our proposed model. In this task, subjects were presented with a series of two-character Chinese words, including real words and pseudowords, and were instructed to press a key when a real word occurred (Go trial) and to stop key press when a pseudoword occurred (No-Go trial). The Go and No-Go trials correspond to the response and inhibition states, respectively.

Data were collected from 20 subjects, each completing four sessions. Each session contained 150 trials, including 120 Go trials and 30 No-Go trials. The trials were arranged in a designed order, with a No-Go trial occurring after a variable-length sequence of Go trials.

4.1.1 Data Preprocessing and Transformation

We first applied standard preprocessing to the fMRI data using SPM12 under MATLAB, including slice timing correction, motion correction, spatial normalization, and spatial smoothing. Since event-related fMRI data cannot be directly used as input in trial-based brain state classification, we transformed the preprocessed fMRI time series into beta series using the Least Squares All (LSA) method [19], applying a General Linear Model (GLM) to estimate trial-wise beta values. Specifically, after this transformation, each trial corresponds to a 3D beta map composed of beta values, similar to how each time point corresponds to a 3D fMRI volume. As a result, each session produced 120 beta maps for Go trials and 30 beta maps for No-Go trials.

4.1.2 Data Cleaning and Balancing

To ensure data quality, we removed the error trials, including error Go trials and error No-Go trials. Error Go trials refer to trials in which subjects failed to press the key when a real word occurred, whereas error No-Go trials refer to trials in which subjects pressed the key when a pseudoword occurred.

Since Go trials were more frequent than No-Go trials, we then performed data balancing by downsampling the beta maps of Go trials to achieve a 1:1 ratio. Specifically, we retained 24 Go and 24 No-Go beta maps for each session, as the accuracy of the 30 No-Go trials in most sessions is above 80 %, and the accuracy of the 120 Go trials in most sessions is above 90 %. For sessions with more correct No-Go trials, we only selected a subset of them, whereas for sessions with insufficient correct No-Go trials, we resampled the correct trials to ensure that the number of beta maps was consistent across all sessions and all subjects. The Go trials we selected were the ones immediately preceding the No-Go trials, as choosing these trials better avoids subjects predicting whether to press the key based on the pattern of Go and No-Go trial occurrences, and instead ensures that their decision to press the key is based on the word presented.

4.1.3 Graph Construction

We then constructed a graph dataset from the beta maps, using the Brainnetome Atlas [20] to define regions at multiple hierarchical levels, and extracting average beta series from each region. Specifically, there are 246 brain regions at the subregion level, 48 at the gyrus level, and 14 at the lobe level, with the number of regions in the left and right hemispheres being half of these amounts, respectively.

Since graph construction requires sequential data to compute connections, a single beta map cannot be used to construct a graph. Therefore, we concatenated beta maps from multiple trials with the same label (response or inhibition) to obtain a 4D beta map, and then constructed the graph in the same way as for 4D fMRI data, as detailed in Section 3.1. Specifically, we concatenated 5 beta maps to form one

sample, applying a sliding window with a stride of 2 on the balanced data, producing 10 samples per session for each label. Thus, each subject contributed 40 response samples from the Go trials and 40 inhibition samples from the No-Go trials, resulting in a total of 800 response samples and 800 inhibition samples across the entire dataset, with 1 600 samples in total.

4.2 Experimental Setup

4.2.1 Model Implementation

We implemented HiBiGNN using the PyTorch library in a Python environment, with PyTorch Geometric utilized for efficient graph data processing and graph neural network operations. The HiBiGNN model consists of two HiBiG-Conv layers and two HiBiG-Readout layers for feature extraction, followed by an MLP module for the final graph classification. The output dimension of HiBiG-Conv layers is 32, and the MLP includes two fully connected layers, each with a hidden dimension of 32.

4.2.2 Dataset Splitting and Cross-Validation

To ensure reliable evaluation, the dataset in Section 4.1 was split into 5 folds for cross-validation. The split was based on subject, with each fold containing data from 4 subjects (320 graphs). Specifically, in each iteration of cross-validation, 1 fold was used as the test set, and the remaining 4 folds were further divided, with 3 folds forming the training set and 1 fold forming the validation set. This setup ensured that the overall ratio of training, validation, and test sets remained $3:1:1$ across all iterations, resulting in 960 graphs for training, 320 for validation, and 320 for testing, with a balanced $1:1$ ratio of response and inhibition samples in all sets.

After obtaining the split dataset in each iteration of cross-validation, we repeated the training and evaluation process using 5 different random seeds for each data split to ensure the reliability of the results, yielding a total of 25 runs (5 data splits $\times$ 5 random seeds). The performance metrics of each model in the subsequent comparison and ablation studies are based on the averages of these 25 runs.

4.2.3 Training Settings

The batch size was set to 32, and the maximum number of training epochs was set to 50, with early stopping applied if validation accuracy did not improve for 20 consecutive epochs. The model with the highest validation accuracy was saved and evaluated on the test set. We used the Adam optimizer for model optimization, setting the learning rate to 5e$-$3 and the weight decay to 1e$-$3. In addition, we used the StepLR scheduler to adjust the learning rate, with a step size of 5 epochs and a gamma of 0.5, which helps the model converge more effectively.

4.3 Comparison with Baseline Models

Input	Type	Method	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Traditional Graph	Generic GNNs	GCN	86.39 (6.56)	85.79 (9.21)	89.10 (5.49)	87.00 (5.41)
		GIN	76.65 (5.53)	77.58 (7.74)	77.23 (7.75)	76.84 (4.79)
		GAT	87.00 (5.11)	87.89 (8.33)	87.48 (6.06)	87.22 (4.22)
		GraphSAGE	90.43 (4.80)	90.63 (7.78)	91.33 (5.99)	90.61 (4.24)
	GNNs for fMRI	BrainGNN	59.46 (3.90)	60.27 (4.62)	58.70 (11.95)	58.61 (6.17)
		FBNetGen	60.89 (4.14)	62.10 (6.19)	62.10 (13.48)	60.73 (6.01)
HiBiG (Homo)	Generic GNNs	GCN	83.68 (6.21)	83.19 (8.83)	86.28 (4.68)	84.33 (4.93)
		GIN	83.38 (7.02)	83.06 (9.80)	86.28 (4.44)	84.18 (5.30)
		GAT	86.83 (4.60)	87.38 (8.06)	87.65 (5.59)	87.07 (3.69)
		GraphSAGE	90.49 (3.66)	91.49 (7.02)	90.33 (6.04)	90.52 (3.26)
HiBiG	Ours	HiBiGNN-GCN	94.68 (4.19)	93.97 (6.85)	96.15 (2.38)	94.89 (3.71)
		HiBiGNN-GIN	85.33 (6.89)	85.57 (10.34)	87.70 (5.74)	86.00 (5.27)
		HiBiGNN-GAT	94.75 (4.49)	93.96 (7.02)	**96.35** (2.18)	94.98 (3.97)
		HiBiGNN-SAGE	**94.84** (4.17)	**94.34** (6.92)	96.10 (2.53)	**95.04** (3.71)

Table 1. Performance comparison with baseline models

We compared our HiBiGNN with multiple baseline models, including generic GNNs and fMRI-specific GNNs, to evaluate the overall performance of HiBiGNN. The generic GNNs included GCN, GIN, GAT, and GraphSAGE, each consisting of two convolutional layers with an output dimension of 32 and two readout layers using a concatenation of mean pooling and max pooling, consistent with the settings of our HiBiGNN. The fMRI-specific GNNs included two popular open-source models:

1. BrainGNN, whose structure was consistent with the original implementation, and

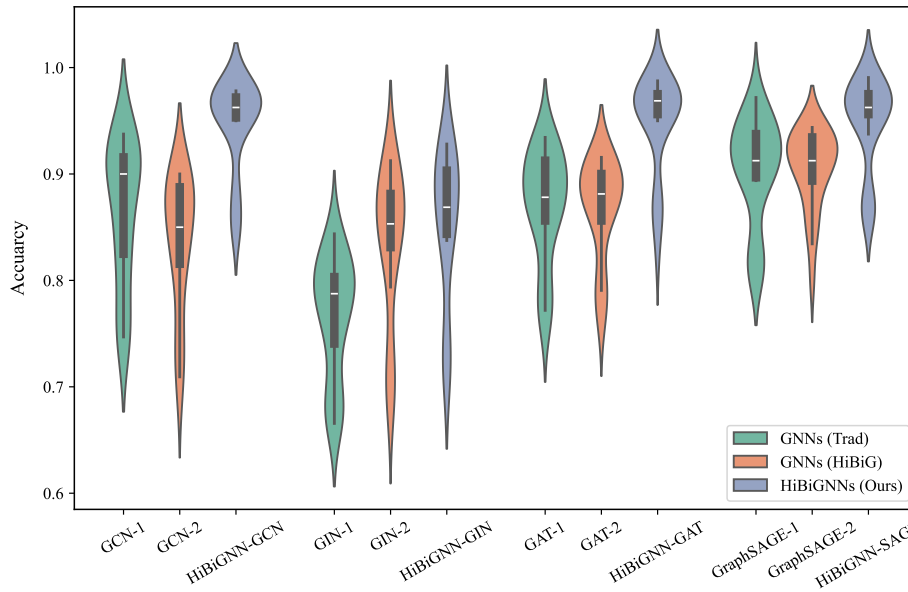


Figure 4. Comparison of models within convolutional method groups on accuracy. Models are grouped into four categories based on convolutional methods (GCN, GIN, GAT, GraphSAGE), within each of which the accuracies of three models over 25 runs are compared: (1) *GNNs (Trad)* – generic GNNs with traditional graph structure; (2) *GNNs (HiBiG)* – generic GNNs with HiBiG (Homo); (3) *HiBiGNNs (Ours)* – our proposed models.

2. FBNetGen, which employed a version with Bi-GRU as the time series encoder.

The comparison results are summarized in Table 1, reflecting the average scores and standard deviations of four metrics from 25 runs described in Section 4.2.2, with accuracy, precision, recall, and F1-score as evaluation metrics. A more intuitive comparison of accuracy is shown in Figure 4, which presents the comparison of models within four convolutional method groups.

In particular, we considered the impact of the input graph structure. Traditional graph structure used in previous fMRI-based studies employed only a single level (subregion-level) of brain regions as nodes, whereas the proposed HiBiG structure employs multi-level brain regions, resulting in more nodes and different connectivity patterns. If baseline models are compared using only traditional graph structure as input, whereas HiBiGNN uses HiBiG, the observed performance differences could result from either the graph structure or the model architecture. To ensure a fair evaluation, we transformed the heterogeneous HiBiG into homogeneous graph structure, denoted as HiBiG (Homo), preserving all nodes and edges though converting them into a single type. Specifically, both HiBiG and HiBiG (Homo) use all three

levels (subregion-level, gyrus-level, and lobe-level) of brain regions, and a discussion on the number of hierarchical levels is provided in Section 4.4.1.

In our experiment, generic GNN models use both the traditional graph structure and HiBiG (Homo) as input, whereas fMRI-specific GNN models use only the traditional graph structure, as they are incompatible with the HiBiG (Homo) structure. In contrast, our proposed HiBiGNN model use the heterogeneous HiBiG as input. Since HiBiGNN can be instantiated with different graph convolution operations, we applied those from GCN, GIN, GAT, and GraphSAGE to create different HiBiGNN instances, denoted as HiBiGNN-GCN, HiBiGNN-GIN, HiBiGNN-GAT, and HiBiGNN-SAGE, respectively.

Notably, the convolution operations of GIN, GAT, and GraphSAGE support message passing between different types of source and target nodes, whereas GCN does not inherently support this functionality. Inspired by R-GCN, we modified the GCN-Conv operation, replacing the degree normalization with a simple averaging operation to support message passing in heterogeneous graphs.

The results in Table 1 and Figure 4 indicate that all HiBiGNN instances outperform their corresponding generic GNN models, whether the input of the generic GNN models is traditional graph structure or HiBiG (Homo). In particular, HiBiGNN-GCN, HiBiGNN-GAT and HiBiGNN-SAGE surpass all baseline models, with an accuracy improvement of over 4 %. The two fMRI-specific models (BrainGNN and FBNetGen) perform poorly on this task, which may be due to their use of node correlations from FC as node features, whereas other models (generic GNNs and HiBiGNNs) use data series as node features. Since we constructed graph samples using short data series, more information is embedded in the series themselves rather than in the correlations calculated from them. Therefore, using data series themselves as features is more effective.

It is noteworthy that HiBiGNN-SAGE and GraphSAGE achieve the highest accuracy within their corresponding groups (HiBiGNNs and generic GNNs), whereas HiBiGNN-GIN and GIN show the lowest accuracy. Specifically, HiBiGNN-GIN and GIN with traditional graph structure perform considerably worse than other models in their respective groups, with accuracy declining by more than 9 %, suggesting that GIN is much less effective in extracting relevant features for this task.

To assess the impact of input graph structures, we compared generic GNNs using traditional graph structure with those using HiBiG (Homo) as input. However, the results in Table 1 and Figure 4 indicate that, on its own, the use of HiBiG (Homo) instead of traditional graph structure does not lead to consistent improvements or declines. Specifically, when using HiBiG (Homo), GIN show a substantial improvement, GCN show a slight decrease, whereas GAT and GraphSAGE show minimal change.

In contrast, pairing HiBiG with HiBiGNNs rather than generic GNNs leads to improvements across all metrics, highlighting HiBiGNNs' ability to better leverage the rich information in HiBiG. Similarly, it can be concluded that complex input structures, such as HiBiG, require specialized models like HiBiGNN to unlock their full potential.

4.4 Ablation Study

In this section, we conducted comprehensive experiments to evaluate the impact of the HiBiG structure and the components of HiBiGNN.

4.4.1 Convolutional Layer and Input Graph Structure

Input	Method	Levels	Bilateral	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Traditional Graph	Graph-SAGE	1		90.49 (3.66)	91.49 (7.02)	90.33 (6.04)	90.52 (3.26)
HiG	HiGNN-SAGE	2		92.34 (3.57)	92.16 (5.72)	93.10 (4.55)	92.44 (3.25)
		3		93.63 (2.87)	93.37 (4.76)	94.25 (3.35)	93.70 (2.70)
BiG	BiGNN-SAGE	1	✓	93.49 (4.07)	94.05 (6.80)	93.53 (3.23)	93.61 (3.65)
HiBiG	HiBiGNN-SAGE	2	✓	94.21 (4.52)	93.73 (7.46)	95.63 (3.07)	94.45 (3.97)
		3	✓	**94.84 (4.17)**	**94.34 (6.92)**	**96.10 (2.53)**	**95.04 (3.71)**

Table 2. Ablation experiments on convolutional layer

The proposed HiBiG incorporates both hierarchical and bilateral structures, and the design of the HiBiG-Conv layer in HiBiGNN is closely related to this approach. Therefore, the impacts of the two structures should be examined separately. Since the comparison experiments revealed that HiBiGNN-SAGE performed the best, we subsequently conducted ablation using models with SAGE-Conv.

Specifically, we compared the performance of the following models: GraphSAGE with traditional graph structure (single-level, no hierarchical or bilateral structure), HiGNN-SAGE with HiG (multi-level, hierarchical structure only), BiGNN-SAGE with BiG (single-level, bilateral structure only), and HiBiGNN-SAGE with HiBiG (multi-level, both hierarchical and bilateral structures). We also explored the effect of adding hierarchical levels by sequentially incorporating the gyrus-level and lobe-level regions into the subregion-level graph.

As shown in Table 2, both HiGNN-SAGE and BiGNN-SAGE outperform the baseline GraphSAGE, and HiBiGNN-SAGE perform the best. This demonstrates that both hierarchical and bilateral designs are effective in this task, and their combination is more effective than either alone.

Furthermore, we also found that increasing hierarchical levels can improve model performance. Specifically, a generic GNN can be viewed as a single-level HiGNN, and a BiGNN can be viewed as a single-level HiBiGNN. In this perspective, both

HiGNN and HiBiGNN show a gradual improvement in accuracy and other metrics as the number of hierarchical levels increased from 1 to 3.

4.4.2 Readout Layer

Type	Method	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Substitution of Global Pooling	Mean Pooling Only	93.39 (6.07)	92.48 (9.21)	95.88 (3.43)	93.83 (5.14)
	Max Pooling Only	94.53 (3.78)	94.48 (5.98)	95.10 (4.14)	94.62 (3.52)
Substitution of Subgraph Aggregation	Average for Both Hemi and Level	94.15 (4.32)	93.61 (6.58)	95.35 (3.20)	94.33 (3.92)
	Average for Hemi and Concat for Level	94.55 (4.21)	94.17 (6.56)	95.58 (3.72)	94.70 (3.87)
Standard Method	HiBiG-Readout	**94.84** (4.17)	**94.34** (6.92)	**96.10** (2.53)	**95.04** (3.71)

Table 3. Ablation experiments on readout layer

The computation in the proposed HiBiG-Readout layer includes three steps: global pooling of hemispheric subgraphs, aggregation of the left and right hemispheric subgraph features, and aggregation of hierarchical subgraph features. The global pooling step employs both mean pooling and max pooling, the hemispheric aggregation step employs weighted average, and the hierarchical aggregation step employs concatenation.

To assess the impact of the global pooling and aggregation methods, we replaced them with alternative methods and compared the results to the proposed standard model.

As shown in Table 3, for the global pooling step, using both mean pooling and max pooling (as in our standard model) yields better results than using one alone, and max pooling makes a greater contribution than mean pooling in this task. For the aggregation steps, the hemispheric aggregation using weighted average outperforms the one using simple average (denoted as 'average'), the hierarchical aggregation using concatenation (denoted as 'concat') performs better than using average, and our standard model adopts the optimal methods in these steps.

Notably, for hierarchical aggregation, if the subgraph feature sizes of different levels are not the same, additive aggregation (such as simple or weighted average) cannot be used. As a result, concatenation offers better versatility in such scenarios. However, for hemispheric aggregation, concatenation is unnatural since the left and right subgraphs are simply splits of a traditional graph, and additive aggregation can maintain consistency with the graph feature size extracted from a traditional graph by generic GNNs, so we did not consider using concatenation for hemispheric aggregation.

4.4.3 Inter-Hemispheric Connectivity

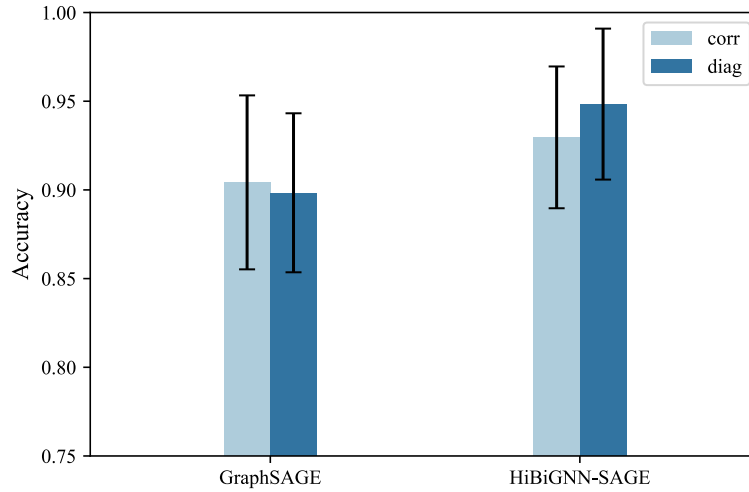


Figure 5. Comparison of inter-hemispheric connectivity methods on accuracy. Two connectivity methods are compared: 1) corr – brain region signal correlation, and 2) diag – diagonal matrix which connects corresponding regions in the left and right hemispheres.

In the proposed HiBiG structure, the connections between the left and right hemispheric subgraphs are defined using a diagonal matrix, with connections existing only between corresponding left and right brain regions. Whereas in the traditional graph structure, there is no distinction between the left and right brain regions, and all connections are derived from the correlation matrix.

We assessed the impact of the two inter-hemispheric connectivity method. For both traditional graph structure and HiBiG, we constructed the inter-hemispheric connections using both a diagonal matrix and a correlation matrix (denoted as 'diag' and 'corr').

To assess the impact of the two inter-hemispheric connectivity methods, we constructed the inter-hemispheric connections for both the traditional graph structure and HiBiG using a diagonal matrix and a correlation matrix (denoted as 'diag' and 'corr'). We then compared the performance of GraphSAGE with the two types of traditional graphs and the performance of HiBiGNN-SAGE with the two types of HiBiGs.

As shown in Figure 5, the performance of GraphSAGE models with traditional graph structure is comparable between the two connectivity methods, whereas the performance of HiBiGNN models with HiBiG is evidently better with 'diag' connectivity than with 'corr' connectivity, which indicates that the 'diag' connectivity is more useful for HiBiGNN with HiBiG, supporting better information exchange between the left and right hemispheric subgraphs.

4.5 Visualization of t-SNE Results

To compare the feature extraction ability of different models, we used t-SNE to visualize the input graph data as well as the features extracted by HiBiGNNs and generic GNNs from both the training and test sets.

To prepare the input features for t-SNE, since graph data cannot be directly used by t-SNE, we concatenated the feature vectors of all nodes, and visualized the resulting vector. For the models, we visualized the graph-level feature obtained after the readout operation, which serves as the input to the classification MLP.

As shown in Figure 6, regardless of whether the input graph data have a single-level traditional graph structure or a multi-level HiBiG structure, the two classes are not well separated in the 2D projection space.

Among generic GNNs, GraphSAGE extracts features with a clearer boundary between the two classes compared to other models. This is consistent with the results in Table 1, where GraphSAGE outperforms the other generic GNNs on accuracy and other metrics.

Among HiBiGNNs, all instances except HiBiGNN-GIN extract features that distinctly separate the two classes, outperforming generic GNNs in this regard and demonstrating good generalization on the test set, with HiBiGNN-SAGE showing the clearest separation between the classes. In contrast, HiBiGNN-GIN fails to achieve a clear separation, which is consistent with its lower accuracy compared to the other HiBiGNN instances.

5 DISCUSSION

5.1 Implications and Applications

This study highlights the potential of HiBiGNN in improving fMRI-based brain network analysis by incorporating both hierarchical and bilateral structures. It is worth noting that HiBiGNN is a generic architecture that can be instantiated with different graph convolution operations, providing greater flexibility for adapting to a wider range of datasets.

Since the input HiBiG of HiBiGNN differs from traditional graphs only in structure, and the data for HiBiG construction are still average time series of brain regions, HiBiGNN can be naturally applied to various graph classification tasks from previous studies, such as gender classification, cognitive state classification, and disease prediction. Therefore, similar to previous models, HiBiGNN can also bring benefits to neuroscience, psychology, clinical medicine, and other fields.

5.2 Limitations and Future Work

Despite its advantages, HiBiGNN has several limitations. One key limitation is that the process of graph construction and feature extraction is not end-to-end. Consequently, the quality of the constructed graph data may significantly influence the

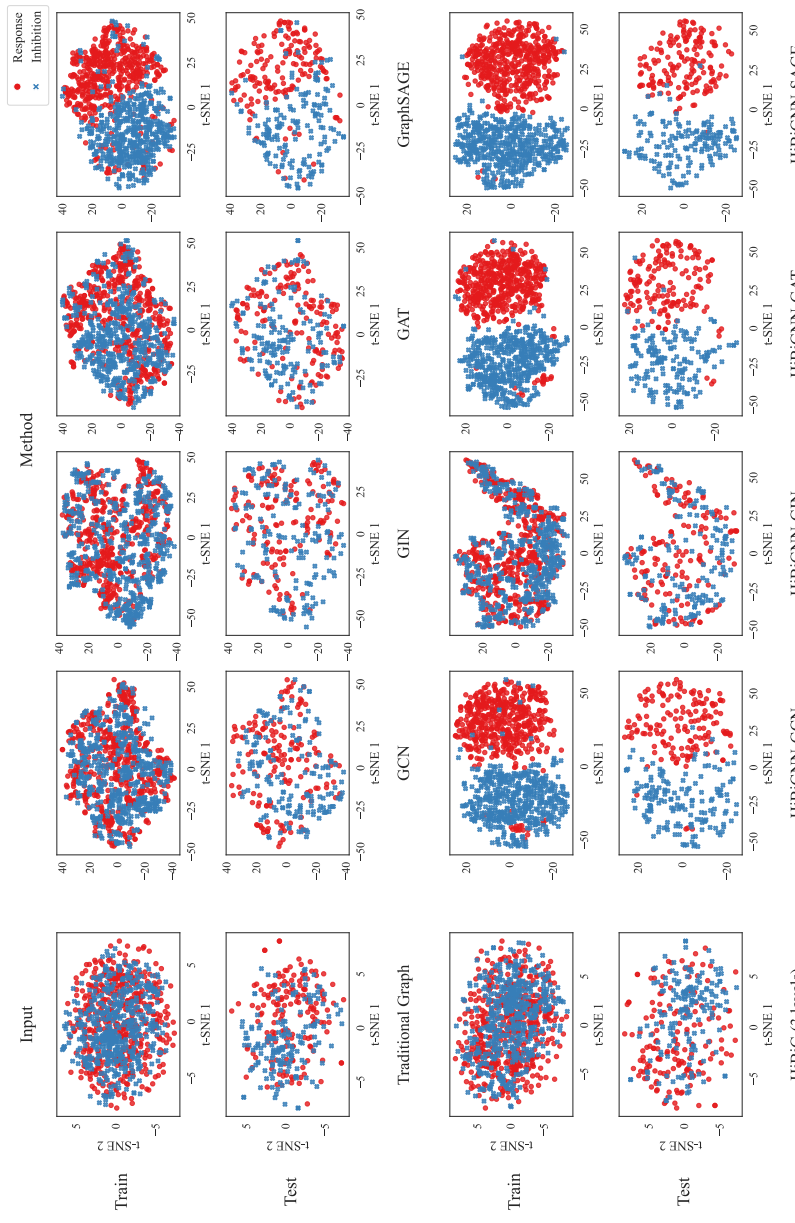


Figure 6. Comparison of t-SNE embeddings for models with different convolutional methods. The upper two rows show the t-SNE results of generic GNNs with traditional graphs as input, and the lower two rows show the t-SNE results of HiBiGNNs with HiBiG as input.

performance of the model. Future work could explore methods for incorporating the graph construction process into the training pipeline, enabling the generation of connections between nodes through learnable methods, and automatically identifying multi-level brain regions to overcome the constraints of predefined brain atlases.

Another limitation is that, due to the multiple types of nodes and relations in heterogeneous graphs, HiBiGNN cannot directly apply node selection methods used in homogeneous graphs, such as top-k pooling, to preserve important nodes for graph classification tasks. The diversity of nodes and edges necessitates tailored node selection techniques for effective feature extraction. Future work could focus on developing specialized node selection methods tailored to the HiBiG structure, enabling HiBiGNN to preserve important nodes when handling the diversity of nodes and relations.

The interpretability of HiBiGNN is also a challenge, as HiBiGNN does not inherently compute node-level importance, making it difficult to analyze the contributions of different brain regions to the model's prediction. Future work could integrate attention mechanisms or techniques for improving interpretability to enhance the understanding of learned representations, thereby facilitating neuroscientific insights.

Additionaly, extending HiBiG and HiBiGNN to othter brain imaging modalities, such as electroencephalography (EEG), is a promising research direction. Similar to fMRI, EEG data can also be used to construct brain networks, with nodes representing electrodes and edges defined by signal correlations. Consequently, we can aggregate EEG electrodes into coarser regions to create multi-level brain nodes, and then assign these nodes to hemispheric subgraphs based on their location, which yields HiBiG and enables the application of HiBiGNN to EEG data.

6 CONCLUSION

In this paper, we propose the novel HiBiG structure and the innovative HiBiGNN architecture to address the underutilization of hierarchical and bilateral information in fMRI-based brain network analysis. Evaluation on a self-collected fMRI dataset from a response inhibition task demonstrates the superiority of HiBiGNN over several generic and fMRI-specific GNN models, emphasizing the importance of considering both hierarchical and bilateral aspects in brain network analysis.

Future research could focus on how to automatically identify multi-level brain regions without relying on predefined brain atlases, and how to construct inter-hemispheric and inter-level connections in HiBiG using learnable methods. Furthermore, we believe that the hierarchical and bilateral structures are not just beneficial for fMRI analysis, and it is also a promising direction for extending HiBiG and HiBiGNN to other modalities of brain imaging data for further exploration.

Funding

This work was supported by the Special Project of Clinical Research in Health Industry of the Shanghai Municipal Health Commission (Grant No. 20224Y0207), the Research Project of China Disabled Persons' Federation – on assistive technology (Grant No. 2023CDPFAT-12), the National Natural Science Foundation of China (Grant No. 62472319), the Second Round of "Three-Year Action Plan to Promote Clinical Skills and Clinical Innovation in Municipal Hospitals" Research Physician Innovation Transformation Capability Training Program (Grant No. SHDC2023CRT001), and the Shanghai Research Center of Rehabilitation Medicine (Top Priority Research Center of Shanghai) (Grant No. 2023ZZ02027).

REFERENCES

[1] Kipf, T. N.—Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. 5th International Conference on Learning Representations (ICLR 2017), 2017, doi: 10.48550/arXiv.1609.02907.

[2] Hamilton, W.—Ying, Z.—Leskovec, J.: Inductive Representation Learning on Large Graphs. In: Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 30 (NIPS 2017). Curran Associates, Inc., 2017, pp. 1024–1034, doi: 10.48550/arXiv.1706.02216.

[3] Veličković, P.—Cucurull, G.—Casanova, A.—Romero, A.—Liò, P.—Bengio, Y.: Graph Attention Networks. 6th International Conference on Learning Representations (ICLR 2018), 2018, doi: 10.48550/arXiv.1710.10903.

[4] Xu, K.—Hu, W.—Leskovec, J.—Jegelka, S.: How Powerful Are Graph Neural Networks? 7th International Conference on Learning Representations (ICLR 2019), 2019, doi: 10.48550/arXiv.1810.00826.

[5] Yan, Y.—Zhu, J.—Duda, M.—Solarz, E.—Sripada, C.—Koutra, D.: GroupINN: Grouping-Based Interpretable Neural Network for Classification of Limited, Noisy Brain Data. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19), 2019, pp. 772–782, doi: 10.1145/3292500.3330921.

[6] Li, X.—Zhou, Y.—Dvornek, N.—Zhang, M.—Gao, S.—Zhuang, J.—Scheinost, D.—Staib, L. H.—Ventola, P.—Duncan, J. S.: BrainGNN: Interpretable Brain Graph Neural Network for fMRI Analysis. Medical Image Analysis, Vol. 74, 2021, Art. No. 102233, doi: 10.1016/j.media.2021.102233.

[7] Cui, H.—Dai, W.—Zhu, Y.—Li, X.—He, L.—Yang, C.: Interpretable Graph Neural Networks for Connectome-Based Brain Disorder Analysis. In: Wang, L., Dou, Q., Fletcher, P. T., Speidel, S., Li, S. (Eds.): Medical Image Computing and Computer Assisted Intervention – MICCAI 2022. Springer, Cham, Lecture Notes in Computer Science, Vol. 13438, 2022, pp. 375–385, doi: 10.1007/978-3-031-16452-1_36.

[8] Kan, X.—Cui, H.—Lukemire, J.—Guo, Y.—Yang, C.: FBNETGEN: Task-Aware GNN-Based fMRI Analysis via Functional Brain Network Generation. In:

Konukoglu, E., Menze, B., Venkataraman, A., Baumgartner, C., Dou, Q., Albarqouni, S. (Eds.): Proceedings of the 5th International Conference on Medical Imaging with Deep Learning. Proceedings of Machine Learning Research (PMLR), Vol. 172, 2022, pp. 618–637, doi: 10.48550/arXiv.2205.12465.

[9] Kan, X.—Dai, W.—Cui, H.—Zhang, Z.—Guo, Y.—Yang, C.: Brain Network Transformer. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.): Advances in Neural Information Processing Systems 35 (NeurIPS 2022). Curran Associates, Inc., 2022, pp. 25586–25599, doi: 10.48550/arXiv.2210.06681.

[10] Kim, B. H.—Ye, J. C.—Kim, J. J.: Learning Dynamic Graph Representation of Brain Connectome with Spatio-Temporal Attention. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. S., Wortman Vaughan, J. (Eds.): Advances in Neural Information Processing Systems 34 (NeurIPS 2021). Curran Associates, Inc., Vol. 34, 2021, pp. 4314–4327, doi: 10.48550/arXiv.2105.13495.

[11] Yan, J.—Chen, Y.—Yang, S.—Zhang, S.—Jiang, M.—Zhao, Z.—Zhang, T.—Zhao, Y.—Becker, B.—Liu, T.—Kendrick, K.—Jiang, X.: Multi-Head GAGNN: A Multi-Head Guided Attention Graph Neural Network for Modeling Spatio-Temporal Patterns of Holistic Brain Functional Networks. In: de Bruijne, M., Cattin, P. C., Cotin, S., Padoy, N., Speidel, S., Zheng, Y., Essert, C. (Eds.): Medical Image Computing and Computer Assisted Intervention – MICCAI 2021. Springer, Cham, Lecture Notes in Computer Science, Vol. 12907, 2021, pp. 564–573, doi: 10.1007/978-3-030-87234-2_53.

[12] Liu, L.—Wen, G.—Cao, P.—Hong, T.—Yang, J.—Zhang, X.—Zaiane, O. R.: BrainTGL: A Dynamic Graph Representation Learning Model for Brain Network Analysis. Computers in Biology and Medicine, Vol. 153, 2023, Art. No. 106521, doi: 10.1016/j.compbiomed.2022.106521.

[13] Schlichtkrull, M.—Kipf, T. N.—Bloem, P.—van den Berg, R.—Titov, I.—Welling, M.: Modeling Relational Data with Graph Convolutional Networks. In: Gangemi, A., Navigli, R., Vidal, M. E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M. (Eds.): The Semantic Web (ESWC 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 10843, 2018, pp. 593–607, doi: 10.1007/978-3-319-93417-4_38.

[14] Zhang, C.—Song, D.—Huang, C.—Swami, A.—Chawla, N. V.: Heterogeneous Graph Neural Network. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19), 2019, pp. 793–803, doi: 10.1145/3292500.3330961.

[15] Hu, Z.—Dong, Y.—Wang, K.—Sun, Y.: Heterogeneous Graph Transformer. Proceedings of The Web Conference 2020 (WWW '20), 2020, pp. 2704–2710, doi: 10.1145/3366423.3380027.

[16] Wang, X.—Ji, H.—Shi, C.—Wang, B.—Ye, Y.—Cui, P.—Yu, P. S.: Heterogeneous Graph Attention Network. The World Wide Web Conference (WWW '19), 2019, pp. 2022–2032, doi: 10.1145/3308558.3313562.

[17] Fu, X.—Zhang, J.—Meng, Z.—King, I.: MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. Proceedings of The Web Conference 2020 (WWW '20), 2020, pp. 2331–2341, doi: 10.1145/3366423.3380297.

[18] Yun, S.—Jeong, M.—Kim, R.—Kang, J.—Kim, H. J.: Graph Transformer

Networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 32 (NeurIPS 2019). Curran Associates, Inc., 2019, pp. 11983–11993, doi: 10.48550/arXiv.1911.06455.

[19] Rissman, J.—Gazzaley, A.—D'Esposito, M.: Measuring Functional Connectivity During Distinct Stages of a Cognitive Task. NeuroImage, Vol. 23, 2004, No. 2, pp. 752–763, doi: 10.1016/j.neuroimage.2004.06.035.

[20] Fan, L.—Li, H.—Zhuo, J.—Zhang, Y.—Wang, J.—Chen, L.—Yang, Z.—Chu, C.—Xie, S.—Laird, A. R.—Fox, P. T.—Eickhoff, S. B.—Yu, C.—Jiang, T.: The Human Brainnetome Atlas: A New Brain Atlas Based on Connectional Architecture. Cerebral Cortex, Vol. 26, 2016, No. 8, pp. 3508–3526, doi: 10.1093/cercor/bhw157.

Zhengyuan Fan received his B.Sc. degree in computer science from the Tongji University in 2022. Now, he is a graduate student in the School of Computer Science and Technology, Tongji University. His research interests include machine learning, brain computer interface, and cognitive neuroscience.

Hongfei Ji is Associate Professor at the School of Computer Science and Technology, Tongji University, Shanghai, China. He received his Ph.D. degree in software engineering from the Tongji University, Shanghai, China, in 2015. His research interests include machine learning, pattern recognition, brain computer interface, and cognitive neuroscience.

Jie Li is Associate Professor in the School of Computer Science and Technology, Tongji University, Shanghai, China. She received her Ph.D. degree in computer science from the Shanghai Jiao Tong University, Shanghai, China, in 2010. Her research interests include machine learning, brain computer interface, brain like computation, and cognitive neuroscience.

Jie Zhuang is Professor at the School of Psychology, Shanghai University of Sport. He received his Ph.D. degree from the University of Cambridge, UK, in 2010. His research primarily focuses on the brain mechanisms underlying language and emotion processing, as well as related neural plasticity, employing an integrated approach that includes functional magnetic resonance imaging (fMRI), transcranial electromagnetic stimulation (TMS, TI, tDCS, tACS), and electroencephalography (EEG/ERP).

Qian Qian is Associate Chief Speech-Language Pathologist at the Neurorehabilitation Center, Yangzhi Affiliated Rehabilitation Hospital of Tongji University (Shanghai Sunshine Rehabilitation Center), Shanghai, China. Her research interests include the neural mechanisms and intervention strategies of post-stroke aphasia.