BrainGNN: Interpretable Brain Graph Neural Network for fMRI Analysis

Xiaoxiao Li[1⋆], Yuan Zhou[4], Siyuan Gao[1⋆⋆], Nicha Dvornek[1,4⋆⋆], Muhan Zhang[6⋆⋆], Juntang Zhuang[1], Shi Gu[5], Dustin Scheinost[4], Lawrence Staib[1,3,4], Pamela Ventola[2], and James Duncan[1,3,4]

[1] Biomedical Engineering, Yale University, New Haven, CT, 06511, USA
[2] Child Study Center, Yale School of Medicine, New Haven, CT, 06511, USA
[3] Electrical Engineering, Yale University, New Haven, CT, 06511, USA
[4] Radiology & Biomedical Imaging, Yale School of Medicine, New Haven, CT, 06511, USA
[5] Department of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[6] Facebook AI

Abstract. Understanding how certain brain regions relate to a specific neurological disorder or cognitive stimuli has been an important area of neuroimaging research. We propose BrainGNN, a graph neural network (GNN) framework to analyze functional magnetic resonance images (fMRI) and discover neurological biomarkers. Considering the special property of brain graphs, we design novel ROI-aware graph convolutional (Ra-GConv) layers that leverage the topological and functional information of fMRI. Motivated by the need for transparency in medical image analysis, our BrainGNN contains ROI-selection pooling layers (R-pool) that highlight salient ROIs (nodes in the graph), so that we can infer which ROIs are important for prediction. Furthermore, we propose regularization terms - unit loss, topK pooling (TPK) loss and group-level consistency (GLC) loss - on pooling results to encourage reasonable ROI-selection and provide flexibility to preserve either individual- or group-level patterns. We apply the BrainGNN framework on two independent fMRI datasets: Autism Spectral Disorder (ASD) fMRI dataset and Human Connectome Project (HCP) 900 Subject Release. We investigate different choices of the hyperparameters and show that BrainGNN outperforms the alternative fMRI image analysis methods in terms of four different evaluation metrics. The obtained community clustering and salient ROI detection results show high correspondence with the previous neuroimaging-derived evidence of biomarkers for ASD and specific task states decoded for HCP.

1 Introduction

The brain is an exceptionally complex system and understanding it's functional organization is the goal of modern neuroscience. Using fMRI, large strides in

⋆ Corresponding Author: Xiaoxiao Li, xiaoxiao.li@aya.yale.edu
⋆⋆ Equal contribution

understanding this organization have been made by modeling the brain as a graph—a mathematical construct describing the connections or interactions (i.e. edges) between different discrete objects (i.e. nodes). To create these graphs, nodes are defined as brain regions of interest (ROIs) and edges are defined as the functional connectivity between those ROIs, computed as the pairwise correlations of functional magnetic resonance imaging (fMRI) time series, as illustrated in Fig. 1.

Traditional graph-based analyses for fMRI have focused on two-stage methods: stage 1 — feature engineering from graphs and stage 2 — analysis on the extracted features. For feature engineering, studies have used graph theoretical metrics to summarize the functional connectivity for each node into statistical measurements [44,22]. Additionally, due to the high dimensionality of fMRI data, usually ROIs are clustered into highly connected communities to reduce dimensionality [32,10] or performing data-driven feature selection [38]. For these two-stage methods, if the results from the first stage are not reliable, significant errors can be induced in the second stage.

The past few years have seen growing prevalence of using graph neural networks (GNN) for end-to-end graph learning applications. GNNs are the state-of-the-art deep learning methods for most graph-structured data analysis problems. They combine node features, edge features, and graph structure by using a neural network to embed node information and pass information through edges in the graph. As such, they can be viewed as a generalization of the traditional convolutional neural networks (CNN) for images. Due to their high performance and interpretability, GNNs have become a widely applied graph analysis method [25,24,47,48,16,33]. Most existing GNNs are built on graphs that do not have correspondence between the nodes of different instances, such as social networks and protein networks. These methods – including the current GNN methods for fMRI analysis – use the same kernel over different nodes, which implicitly assumes brain graphs are translation invariant. However, nodes in the same brain graph have distinct locations and unique identities. Thus, applying the same kernel over all nodes is problematic. In addition, few GNN studies have explored both individual-level and group-level explanations, which are critical in neuroimaging research.

In this work, we propose a graph neural network-based framework for mapping regional and cross-regional functional activation patterns for classification tasks, such as classifying neurodisorder patients versus healthy control subjects and performing cognitive task decoding. Our framework jointly learns ROI clustering and the downstream whole-brain fMRI analysis. This not only reduces preconceived errors, but also learns particular clustering patterns associated with the downstream tasks. Specifically, from estimated model parameters, we can retrieve ROI clustering patterns. Also, our GNN design facilitates model interpretability by regulating intermediate outputs with a novel loss term, which provides the flexibility to choose between individual-level and group-level explanations.

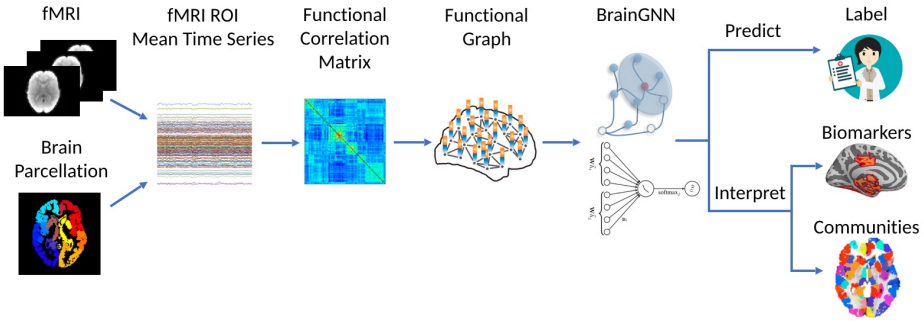


Fig. 1: The overview of the pipeline. fMRI images are parcellated by an atlas and transferred to graphs. Then, the graphs are sent to our proposed BrainGNN, which gives the prediction of specific tasks. Jointly, BrainGNN selects salient brain regions that are informative to the prediction task and clusters brain regions into prediction-related communities.

A preliminary version of this work, *Pooling Regularized Graph Neural Network (PR-GNN) for fMRI Biomarker Analysis* [28] is accepted at the 22st International Conference on Medical Image Computing and Computer Assisted Intervention. This paper extends this work by designing novel graph convolutional layers, justifying the loss terms regarding the pooling layer, and testing our methods on additional datasets.

2 BrainGNN

2.1 Preliminaries

Notation and Problem Definition First we parcellate the brain into N regions of interest (ROIs) based on its T1 structural MRI. We define ROIs as graph nodes $\mathcal{V} = \{v_1, \ldots, v_N\}$ and the nodes are preordered. As brain ROIs can be aligned by brain parcellation atlases based on their location in the structure space, we define the brain graphs as ordered aligned graphs. We define an undirected weighted graph as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ is the edge set, i.e., a collection of (v_i, v_j) linking vertices from v_i to v_j. In our setting, G has an associated node feature set $\mathcal{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_N\}$, where $\mathbf{h}_i$ is the feature vector associated with node v_i. For every edge connecting two nodes, $(v_i, v_j) \in \mathcal{E}$, we have its strength $e_{ij} \in \mathbb{R}$ and $e_{ij} > 0$. We also define $e_{ij} = 0$ for $(v_i, v_j) \notin \mathcal{E}$ and therefore the adjacency matrix $E = [e_{ij}] \in \mathbb{R}^{N \times N}$ is well defined.

Architecture Overview Classification on graphs is achieved by first embedding node features into a low-dimensional space, then grouping nodes and summarizing them. The summarized vector is then fed into a classifier, such as a

multilayer perceptron (MLP), potentially in an end-to-end fashion. Our proposed network architecture is illustrated in Fig. (2a). It is formed by three different types of layers: graph convolutional layers, node pooling layers and a readout layer. Generally speaking, GNNs inductively learn a node representation by recursively transforming and aggregating the feature vectors of its neighboring nodes.

A **graph convolutional layer** is used to probe the graph structure by using edge features, which contain important information about graphs. For example, the weights of the edges in brain fMRI graphs can represent the relationship between different ROIs.

Following [37], we define $\mathbf{h}_i^{(l)} \in \mathbb{R}^{d^{(l)}}$ as the features for the ith node in the lth layer, where $d^{(l)}$ is the dimension of the lth layer features. The propagation model for the forward-pass update of node representation is calculated as:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(W_0^{(l)}\mathbf{h}_i^{(l)} + \sum_{j\in\mathcal{N}^{(l)}(i)} \phi\left(W_1^{(l)}\mathbf{h}_j^{(l)}, e_{ij}^{(l)}\right)\right), \tag{1}$$

where $\mathcal{N}^{(l)}(i)$ denotes the set of indices of neighboring nodes of node v_i, $e_{ij}^{(l)}$ denotes the features associated with the edge from v_i to v_j, $W_0^{(l)}, W_1^{(l)}$ denote the model's parameters to be learned, and ϕ is any linear/nonlinear function that can be applied on the neighboring nodes' feature embedding. σ is the activation function. The first layer is operated on the original graph, i.e. $\mathbf{h}_i^{(0)} = \mathbf{h}_i$, $e_{ij}^{(0)} = e_{ij}$.

A **node pooling** layer is used to reduce the size of the graph, either by grouping the nodes together or pruning the original graph G to a subgraph G_s by keeping some important nodes only. We will focus on the pruning method, as it is more interpretable and can help detect biomarkers.

A **readout** layer is used to summarize the node feature vectors $\{\mathbf{h}_i^{(l)}\}$ into a single vector $\mathbf{z}^{(l)}$ which is finally fed into a classifier for graph classification.

2.2 Layers in BrainGNN

In this section, we provide insights and highlight the innovative design aspects of our proposed BrainGNN architecture.

ROI-aware Graph Convolutional Layer

Overview We propose an ROI-aware graph convolutional layer (Ra-GConv) with two insights. First, when computing the node embedding, we allow Ra-GConv to learn different convolutional kernels conditioned on the ROI (geometric information of the brain), instead of using the same kernel W on all the nodes as it is shown in Eq. (1). Second, we include edge weights for message filtering, as the magnitude of edge weights presents the connection strength between two ROIs. We assume that more closely connected ROIs have a higher impact on each other.

Design We begin by assuming the graphs have additional regional information and the nodes of the same region from different graphs have similar properties. We propose to encode the regional information to the embedding kernel function for the nodes. Given node i's regional information $\mathbf{r}_i$, such as the node's coordinates in a mesh graph, we propose to learn the vectorized embedding kernel $\text{vec}(W_i^{(l)})$ based on $\mathbf{r}_i$ for the lth Ra-GConv layer:

$$\text{vec}(W_i^{(l)}) = f_{MLP}^{(l)}(\mathbf{r}_i) = \Theta_2^{(l)}\text{relu}(\Theta_1^{(l)}\mathbf{r}_i) + \mathbf{b}^{(l)}, \tag{2}$$

where the MLP network with parameters $\{\Theta_1^{(l)}, \Theta_2^{(l)}\}$ maps $\mathbf{r}_i$ to a $d^{(l+1)} \cdot d^{(l)}$ dimensional vector then reshapes the output to a $d^{(l+1)} \times d^{(l)}$ matrix $W_i^{(l)}$ and $\mathbf{b}^{(l)}$ is the bias term in MLP.

Given a brain parcellated into N ROIs, we order the ROIs in the same manner for all the brain graphs. Therefore, the nodes in the graphs of different subjects are aligned. However, the convolutional embedding should be independent of the ordering methods. Given an ROI ordering for all the graphs, we use one-hot encoding to represent the ROI's location information, instead of using coordinates, because the nodes in the brain are aligned well. Specifically, for node v_i, its ROI representation $\mathbf{r}_i$ is a $N-$dimensional vector with 1 in the ith entry and 0 for the other entries. Assume that $\Theta_1^{(l)} = [\boldsymbol{\alpha}_1^{(l)}, \ldots, \boldsymbol{\alpha}_{N^{(l)}}^{(l)}]$, where $N^{(l)}$ is the number of ROIs in the lth layer, $\boldsymbol{\alpha}_i^{(l)} = [\alpha_{i1}^{(l)}, \ldots, \alpha_{iK^{(l)}}^{(l)}]^T \in \mathbb{R}^{K^{(l)}}$, $\forall i \in \{1, \ldots, N^{(l)}\}$, where $K^{(l)}$ can be seen as the number of clustered communities for the $N^{(l)}$ ROIs. Assume $\Theta_2^{(l)} = [\boldsymbol{\beta}_1^{(l)}, \ldots, \boldsymbol{\beta}_{K^{(l)}}^{(l)}]$ with $\boldsymbol{\beta}_j^{(l)} \in \mathbb{R}^{d^{(l+1)} \cdot d^{(l)}}$, $\forall j \in \{1, \ldots, K^{(l)}\}$. Then Eq. (2) can be rewritten as

$$\text{vec}(W_i^{(l)}) = \sum_{j=1}^{K^{(l)}} (\alpha_{ij}^{(l)})^+ \boldsymbol{\beta}_j^{(l)} + \boldsymbol{b}^{(l)}. \tag{3}$$

We can view $\{\boldsymbol{\beta}_j^{(l)} : j = 1, \ldots, K^{(l)}\}$ as a basis and $(\alpha_{ij}^{(l)})^+$ as the coordinates. From another perspective, $(\alpha_{ij}^{(l)})^+$ can been seen as the non-negative assignment score of ROI i to community j. If we train different embedding kernels for different ROIs for the lth layer, the total parameters to be learned will be $N^{(l)}d^{(l)}d^{(l+1)}$. Usually we have $K^{(l)} \ll N^{(l)}$. By Eq. (3), we can reduce the number of learnable parameters to $K^{(l)}d^{(l)}d^{(l+1)} + N^{(l)}K^{(l)}$ parameters, while still assigning a separate embedding kernel for each ROI. The ROIs in the same community will be embedded by the similar kernel so that nodes in different communities are embedded in different ways.

As the graph convolution operations in [15], the node features will be multiplied by the edge weights, so that neighbors connected with stronger edges have a larger influence. The graph convolutional layer using ROI-aware kernels and edge weights for filtering can be written as:

$$\tilde{\mathbf{h}}_i^{(l+1)} = \text{relu}\left(W_i^{(l)}\mathbf{h}_i^{(l)} + \sum_{j \in \mathcal{N}^{(l)}(i)} \tilde{e}_{ij}^{(l)} W_j^{(l)} \mathbf{h}_j^{(l)}\right), \tag{4}$$

To avoid increasing the scale of output features, the edge features need to be normalized, as in GAT [41] and GNN [26]. Due to the aggregation mechanism, we normalize the weights by $\tilde{e}_{ij}^{(l)} = e_{ij}^{(l)} / \sum_{j \in \mathcal{N}^{(l)}(i)} e_{ij}^{(l)}$.

ROI-topK Pooling Layer

Overview To perform graph-level classification, a layer for dimensionality reduction is needed since the number of nodes and the feature dimension per node are both large. Recent findings have shown that some ROIs are more indicative of predicting neurological disorders than the others [21,2], suggesting that they should be kept in the dimensionality reduction step. Therefore the node (ROI) pooling layer (R-pool) is designed to keep the most indicative ROIs while removing *noisy* nodes, thereby reducing the dimensionality of the entire graph.

Design To make sure that down-sampling layers behave idiomatically with respect to different graph sizes and structures, we adopt the approach in [6,13] for reducing graph nodes. The choice of which nodes to drop is determined based on projecting the node features onto a learnable vector $\mathbf{w}^{(l)} \in \mathbb{R}^{d^{(l)}}$. The nodes receiving lower scores will experience less feature retention. We denote $\tilde{H}^{(l+1)} = [\tilde{\mathbf{h}}_1^{(l+1)}, \ldots, \tilde{\mathbf{h}}_{N^{(l)}}^{(l+1)}]^T$, where $N^{(l)}$ is the number of nodes at the lth layer. Fully written out, the operation of this pooling layer (computing a pooled graph, $(\mathcal{V}^{(l+1)}, \mathcal{E}^{(l+1)})$, from an input graph, $(\mathcal{V}^{(l)}, \mathcal{E}^{(l)})$), is expressed as follows:

$$\begin{aligned}
\mathbf{s}^{(l)} &= \tilde{H}^{(l+1)} \mathbf{w}^{(l)} / \|\mathbf{w}^{(l)}\| \\
\tilde{\mathbf{s}}^{(l)} &= (\mathbf{s}^{(l)} - \mu(\mathbf{s}^{(l)})) / \sigma(\mathbf{s}^{(l)}) \\
\mathbf{i} &= \text{top}k(\tilde{\mathbf{s}}^{(l)}, k) \\
H^{(l+1)} &= (\tilde{H}^{(l+1)} \odot \text{sigmoid}(\tilde{\mathbf{s}}^{(l)}))_{\mathbf{i},:} \\
E^{(l+1)} &= E_{\mathbf{i},\mathbf{i}}^{(l)}.
\end{aligned} \tag{5}$$

Here $\| \cdot \|$ is the L_2 norm, μ and σ take the input vector and output the mean and standard deviation of its elements. The notation top$k$ finds the indices corresponding to the largest k elements in score vector $\mathbf{s}$, $\odot$ is (broadcasted) element-wise multiplication, and $(\cdot)_{\mathbf{i},\mathbf{j}}$ is an indexing operation which takes elements at row indices specified by $\mathbf{i}$ and column indices specified by $\mathbf{j}$ (colon denotes all indices). The pooling operation retains sparsity by requiring only a projection, a point-wise multiplication and a slicing into the original features and adjacency matrix. Different from [6], we added element-wise score normalization $\tilde{\mathbf{s}}^{(l)} = (\mathbf{s}^{(l)} - \mu(\mathbf{s}^{(l)}))/\sigma(\mathbf{s}^{(l)})$, which is important for calculating the GLC loss and TPK loss (introduced in Section 2.3).

Readout Layer Lastly, we seek a "flattening" operation to preserve information about the input graph in a fixed-size representation. Concretely, to summarize

the output graph of the lth conv-pool block, $(\mathcal{V}^{(l)}, \mathcal{E}^{(l)})$, we use

$$\mathbf{z}^{(l)} = \text{mean}\, \mathcal{H}^{(l)} \parallel \max \mathcal{H}^{(l)}, \tag{6}$$

where $\mathcal{H}^{(l)} = \{\mathbf{h}_i^{(l)} : i = 1, ..., N^{(l)}\}$, mean and max operate elementwisely, and $\parallel$ denotes concatenation. To retain information of a graph in a vector, we concatenate both mean and max summarization for a more informative graph-level representation. The final summary vector is obtained as the concatenation of all those summaries (i.e. $\mathbf{z} = \mathbf{z}^{(1)} \parallel \mathbf{z}^{(2)} \parallel \cdots \parallel \mathbf{z}^{(L)}$) and it is submitted to a MLP for obtaining final predictions.

Putting Layers Together All in all, the architecture (as shown in Fig. 2a) consists of two kinds of layers. The input is a weighted graph with its node attributes constructed from fMRI. We form a two-layer GNN block starting with ROI-aware node embedding by the proposed Ra-GConv layer in Section 2.2, followed by the proposed R-pool layer in Section 2.2. The whole network sequentially concatenates these GNN blocks, and readout layers are added after each GNN block. The final summary vector concatenates all those summaries, and an MLP is applied after that to give final predictions.

2.3 Loss Functions

The classification loss is the cross entropy loss:

$$L_{ce} = -\frac{1}{M} \sum_{m=1}^{M} \sum_{c=1}^{C} y_{m,c} \log(\hat{y}_{m,c}), \tag{7}$$

where M is the number of instances, C is the number of classes, y_{mc} is the ground truth label and $\hat{y}_{mc}$ is the model output.
Now we describe the novel loss terms designed to regulate the learning process and control the interpretability.

Unit loss As we mentioned in Section 2.2, to avoid the problem of identifiability, we propose unit loss:

$$L_{unit}^{(l)} = (\|\mathbf{w}^{(l)}\|_2 - 1)^2. \tag{8}$$

Group-level consistency loss We propose group-level consistency (GLC) loss to force BrainGNN select similar ROIs in R-pool layer for different input instances. Because for some applications, users may want to find the common patterns/biomarkers for a certain neuro-prediction task.Note that $\tilde{\mathbf{s}}^{(l)}$ in Eq. (5) is computed from the input $H^{(l)}$. Therefore, for different inputs $H^{(l)}$, the selected entries of $\tilde{\mathbf{s}}^{(l)}$ can be very different. Thus, we add GLC loss regularization to force the $\tilde{\mathbf{s}}^{(l)}$ vectors to be similar for different input instances after the first pooling layer [7].

[7] We do not consider deeper layers, because nodes on different graphs may not be aligned in the deeper layers, thus the entries in $\tilde{\mathbf{s}}^{(l)}$ may not be aligned.

Now, we mathematically describe the novel GLC loss. In each training batch, suppose there are M instances, which can be partitioned into C subsets based on the class labels, $\mathcal{I}_c = \{m : m = 1, \ldots, M, y_{m,c} = 1\}$, for $c = 1, \ldots, C$. And $y_{m,c} = 1$ indicates the mth instance belonging to class c. We form the scoring matrix for the instances belonging to class c as $S_c^{(l)} = [\tilde{\mathbf{s}}_i^{(l)} : i \in \mathcal{I}_c]^T \in \mathbb{R}^{M_c \times N}$, where $M_c = |\mathcal{I}_c|$. The GLC loss can be expressed as:

$$L_{GLC} = \sum_{c=1}^{C} \sum_{i,j \in \mathcal{I}_c} \parallel \tilde{\mathbf{s}}_i^{(l)} - \tilde{\mathbf{s}}_j^{(l)} \parallel^2 = 2 \sum_{c=1}^{C} \mathrm{Tr}((S_c^{(l)})^T L_c S_c^{(l)}), \tag{9}$$

where $L_c = D_c - W_c$ is a symmetric positive semidefinite matrix, W_c is a $M_c \times M_c$ matrix with all 1s, D_c is a $M_c \times M_c$ diagonal matrix with M_c as diagonal elements [43].

TopK pooling loss We propose TopK pooling (TPK) loss to encourage reasonable node selection at R-pool layer. In other word, we hope the top k selected indicative ROIs should have significantly different scores than those of the unselected nodes. Ideally, the scores for the selected nodes should be close to 1 and the scores for the unselected nodes should be close to 0. To achieve this, we rank sigmoid($\tilde{\mathbf{s}}_m^{(l)}$) for the mth instance in a descending order, denote it as $\hat{\mathbf{s}}_m^{(l)} = [\hat{s}_{m,1}^{(l)}, \ldots, \hat{s}_{m,N^{(l)}}^{(l)}]$, and apply a constraint to all the M training instances to make the values of $\hat{\mathbf{s}}_m^{(l)}$ more dispersed. In practice, we define TPK loss using binary cross-entropy as:

$$L_{TPK}^{(l)} = -\frac{1}{M} \sum_{m=1}^{M} \frac{1}{N^{(l)}} \big(\sum_{i=1}^{k} \log(\hat{s}_{m,i}^{(l)}) + \sum_{i=1}^{N^{(l)}-k} \log(1 - \hat{s}_{m,i+k}^{(l)}) \big), \tag{10}$$

We show the kernel density estimate plots of normalized node pooling scores (indication of the importance of the nodes) changing over the training epoch in Fig. 3 when $k = \frac{1}{2}N^{(l)}$. It is clear to see that the pooling scores are more dispersed over time, Hence the top 50% selected nodes have significantly higher importance scores than the unselected ones.

Finally, the final loss function is formed as:

$$L_{total} = L_{ce} + \sum_{l=1}^{L} L_{unit}^{(l)} + \sum_{l=1}^{L} \lambda_1 L_{TPK}^{(l)} + \lambda_2 L_{GLC}, \tag{11}$$

where λ's are tunable hyper-parameters, l indicates the lth GNN block and L is the total number of GNN blocks. GLC loss is only calculated based on the pooling scores of the first pooling layer. We do not constrain GLC loss for the deeper pooling layer because the nodes after the first pooling layer in different graphs might be different.

2.4 Interpretation from BrainGNN

Community Detection from Convolutional Layers The important contribution of our proposed ROI-aware convolutional layer is the implied community clustering patterns in the graph. Discovering brain community patterns is critical to understanding co-activation and interaction in the brain. Revisiting Eq. (3) and following [29], α_{ij}^{+} provides the membership of ROI i to community j. The community assignment is soft and overlaid. Hence, we consider region i belongs to community j if $\alpha_{ij} > \mu(\boldsymbol{\alpha}_i^{+})+\sigma(\boldsymbol{\alpha}_i^{+})$. This gives us a collection of community indices indicating region membership $\{\boldsymbol{i}_j \subset \{1, ..., N\} : j = 1, ..., K\}$.

Biomarker Detection from Pooling Layers Without the added TPK loss (Eq. (10)), the significance of the nodes left after pooling cannot be guaranteed. With TPK loss, pooling scores are more dispersed over time, hence the selected nodes have significantly higher importance scores than the unselected ones.

The strength of the GLC loss controls the tradeoff between individual-level interpretation and group-level interpretation. On the one hand, for precision medicine, individual-level biomarkers are desired for planning targeted treatment. On the other hand, group-level biomarkers are essential for understanding the common characteristic patterns associated with the disease. We can tune the coefficient λ_2 to control different levels of interpretation. Large λ_2 encourages selecting similar nodes, while small λ_2 allows various node selection results for different instances.

3 Experiments and Results

3.1 Datasets

Two independent datasets are used: the Biopoint Autism Study Dataset (Biopoint) [42] and the Human Connectome Project (HCP) 900 Subject Release [40]. For the Biopoint dataset, the aim is to classify Autism Spectrum Disorder (ASD) and Healthy Control (HC). For the HCP dataset, like the recent work [45,47,31], the aim is to decode and map cognitive states of the human brain. Thus, we classify 7 task states - gambling, language, motor, relational, social, working memory (WM), emotion, then infer the decoded task-related salient ROIs from interpretation.

Biopoint Dataset The Biopoint Autism Study Dataset [42] contains 72 ASD children and 43 age-matched ($p > 0.124$) and IQ-matched ($p > 0.122$) neurotypical healthy controls (HCs). For the fMRI scans, subjects perform the "biopoint" task, viewing point-light animations of coherent and scrambled biological motion in a block design [21] ($24s$ per block). The fMRI data are preprocessed using the pipeline described in [42]. The first few frames are discarded, resulting in 146 frames for each fMRI sequence.

The Desikan-Killiany [9] atlas is used to parcellate brain images into 84 ROIs. The mean time series for each node is extracted from a random 1/3 of voxels in the ROI (given an atlas) by bootstrapping. We augment the data 30 times, resulting in $2160 = 72 \times 30$ ASD graphs and $1290 = 43 \times 30$ HC graphs separately. Edges are defined by thresholding (top 10% positive) partial correlations to achieve sparse connections [8]. We use Pearson correlation coefficient as node features. Because Pearson correlation and partial correlation are different measures of fMRI connectivity, we aggregate them by using one to build edge connections and the other to build node features. This is also motivated by recent multi-graph fusion works for neuroimaging analysis that aim to capture different brain activity patterns by leveraging different correlation matrices [49,12].Hence, node feature $\mathbf{h}_i^{(0)} \in \mathbb{R}^{84}$.

HCP Dataset For this dataset, we restrict our analyses to those individuals who participated in all nine fMRI conditions (seven tasks, two rests) with full length of scan, whose mean frame-to-frame displacement is less than 0.1 mm and whose maximum frame-to-frame displacement is less than 0.15 mm (n=506; 237 males; ages 22–37). This conservative threshold for exclusion due to motion is used to mitigate the substantial effects of motion on functional connectivity.

We process the HCP fMRI data with standard methods (see [11] for more details) and parcellated into 268 nodes using a whole-brain, functional atlas defined in a separate sample (see [17] for more details). Task functional connectivity is calculated based on the raw task time series: the mean time series of each node pair were used to calculate the Pearson correlation and partial correlation. We define a weighted undirected graph with 268 nodes per individual per task condition resulting in $3542 = 506 \times 7$ graphs in total. The same graph construction method as for the Biopoint data is used. Hence, node feature $\mathbf{h}_i^{(0)} \in \mathbb{R}^{268}$.

3.2 Experimental Setup

We trained and tested the algorithm on Pytorch in the Python environment using a NVIDIA Geforce GTX 1080Ti with 11GB GPU memory. The model architecture was implemented with 2 conv layers and 2 pooling layers as shown in Fig. (2a), with parameter $N = 84, K^{(1)} = K^{(2)} = 8, d^{(0)} = 84, d^{(1)} = 16, d^{(2)} = 16, C = 2$ for the Biopoint dataset and $N = 268, K^{(1)} = K^{(2)} = 8, d^{(0)} = 268, d^{(1)} = 32, d^{(2)} = 32, C = 7$ for HCP dataset. The pooling ratio is 0.5.The motivation of $K = 8$ comes from the eight functional networks defined by Finn et al. [11].

We will discuss the variation of λ_1 and λ_2 in Section 3.3. We first hold an held-out testing set and then randomly split the rest of dataset into five folds based on subjects, which means that the graphs from a single subject can only appear in either the training, validation, testing dataset. Four folds were used as

[8] Due to the over-smoothing effect in the general graph neural [34,5] for densely connected graphs.

training data, and the left-out fold was used for validation. Adam was used as the optimizer. We trained BrainGNN for 100 iterations with an initial learning rate of 0.001 and annealed to half every 20 epochs. Each batch contained 400 graphs for Biopoint data and 200 graphs for HCP data. The weight decay parameter was 0.005.

3.3 Hyperparameter Discussion and Ablation Study

In this section, we want to answer the following questions:

Q1 How does the hyper-parameters (λ) selection affect model's performance?
Q2 Is our proposed Ra-GConv layer better than vanilla graph convolution?

Hyperparameter discussion setup To answer Q1, we tune the coefficients (λ_1 and λ_2) in the loss function in Eq. (11). Reminding our intuitions of designing TPK loss and GLC loss described in Section 2.3. A large coefficient of TPK loss, λ_1, encourages more separable node importance scores for selected and unselected nodes after pooling; and 2) The coefficient of GLC loss, λ_2, control the similarity of the nodes selected by different instances, which can control the level of interpretability between individual-level and group-level. Small λ_2 would result in variant individual-specific patterns, while large λ_2 would force the model to learn common group-level patterns. As task classification on HCP could achieve consistently high accuracy over the parameter variations, we only show the results on the Biopoint validation dataset in Fig. 4 to better examine the effect of model variations.

Ablation study setup To answer Q2, we perform ablation studies to investigate the potential benefits of our proposed ROI-aware graph convolutional mechanism — Ra-GConv (Section 2.2). Specifically, we compare our proposed Ra-GConv layer with the strategy of directly learning embedding kernels W (with ROI-aware setting), which is denoted as 'vanilla-GConv'.

Results We evaluate the best classification accuracy on the validation sets in the 5-fold cross-validation setting.

Hyperparameter discussion: First, we investigate the effects of λ_2 on the accuracy, as a suitable range of λ_2 should be determined in order to not sacrifice model accuracy. In Fig. 4a, λ_1 is fixed to 0. We notice that the results are stable to the variation of λ_2 in the range 0 - 0.5. When $\lambda_2 = 1$, the accuracy drops. The accuracy reaches the peak when $\lambda_2 = 0.1$. The same as the other deep learning models behave, BrainGNN is overparameterized. Without regularization ($\lambda_2 = 0$), the model is easier to overfit to the training set, while larger regularization on consistency might result in underfitting on the training set. Second, we fix $\lambda_1 = 0.1$ and varied λ_2 again. As the results presented in Fig. 4b, the accuracy drops if we increase λ_2 after 0.2, which follows the same trend in Fig. 4a. However, the accuracy under the setting of $\lambda_2 = 0$ is better than that in Fig. 4a. Probably the λ_1 terms can work as regularization and mitigate the overfitting issue. Last,

		SVM	Random Forest	MLP	BrainNetCNN	Li et al	GraphSAGE	PR-GNN	**BrainGNN**
Biopoint	Accuracy (%)	62.60(1.82)	67.10(6.14)	70.20(11.30)	76.90(10.84)	73.40(7.15)	72.50(9.89)	77.10(8.71)	**77.90(10.18)**
	F1 (%)	62.45(1.14)	67.08(8.10)	70.54(8.95)	76.45(11.85)	74.26(8.06)	73.15(15.91)	77.36(12.30)	**78.09(11.64)**
	Recall (%)	63.10(7.46)	66.98(7.40)	71.11(8.12)	78.90(13.65)	74.03(6.54)	71.80(11.50)	78.26(10.28)	**78.95(10.65)**
	Precision (%)	63.25(5.76)	67.19(8.33)	69.98(12.32)	74.70(17.64)	74.49(10.21)	73.60(14.28)	76.50(14.32)	**77.25(12.64)**
	Parameter (k)	3	3	138	1438	16	6	6	41
HCP	Accuracy (%)	90.00(0.00)	90.20(4.15)	67.20(34.40)	90.60(4.04)	90.80(5.66)	89.80(12.51)	91.20(8.28)	**94.40(4.04)***
	F1 (%)	90.20(5.81)	90.14(5.55)	63.49(41.80)	90.96(3.50)	90.63(5.35)	88.60(13.19)	91.09(8.35)	**94.34(3.27)***
	Recall (%)	89.57(8.04)	90.06(7.35)	67.97(41.66)	91.12(4.13)	90.08(4.58)	89.43(12.43)	91.00(8.95)	**94.29(3.73)***
	Precision (%)	90.85(9.35)	90.22(4.77)	62.97(42.47)	90.81(3.27)	91.20(3.32)	87.80(14.02)	91.14(8.52)	**94.40(3.59)***
	Parameter (k)	36	36	713	4547	34	12	12	96

Table 1: Comparison of the classification performance with of different baseline machine models and state-of-the-art deep learning models. Classification accuracy, f1-score, recall and precision of the testing sets (under 5-fold cross-validation setting) are reported in mean (standard deviation) format. The number of trainable parameters of each models are denoted. We bolder the results generated from our proposed BrainGNN. * indicates significantly outperforming ($p < 0.001$ under one tail two-sample t-test) all the alternative methods.

we fix $\lambda_2 = 0.1$ and vary λ_1 from $0 - 0.5$. As the results shown in Fig. 4c, when we increased λ_1 to 0.2 and 0.5, the accuracy slightly dropped.

Ablation study: As the results shown in Fig. 4, we can conclude that Ra-GConv overall outperformed the vanilla-GConv strategy under all the parameter settings. The reason could be better node embedding from multiple embedding kernels in the Ra-GConv layers, as the vanilla-GConv strategy treats ROIs (nodes) identically and used the same kernel for all the ROIs. Hence, we claim that Ra-GConv can better characterize the heterogeneous representations of brain ROIs.

Based on the hyperparameters results of tuning λ_1 and λ_2 on validation dataset, we choose the best setting of $\lambda_1 = \lambda_2 = 0.1$ for the following baseline comparison experiments, for which we report the results on the held-out testing dataset.

3.4 Comparison with Baseline Methods

Through our empirical analysis we aim to answer the following key questions:

Q3 How well does BrainGNN perform compared to non-neural network-based models in terms of multiple classification evaluation metrics?
Q4 How well does BrainGNN perform compared to non-neural network-based models in terms of multiple classification evaluation metrics?
Q5 Is BrainGNN parsimonious? Compared with other neural network-based methods, how many parameters can it reduce?

To answer Q3, we compare our method with traditional machine learning (ML) methods for fMRI analysis, which take vectorized correlation matrices as inputs, with the dimensional of N^2, where N is the number of parcellated ROIs. The ML baseline methods included Random Forest (1000 trees), SVM (RBF kernel), and MLP (2 layers with 20 hidden nodes). To answer Q4, we compare our

method with the state-of-the-art deep learning (DL) method. A variety of deep methods have been applied to brain connectome data, long short-term memory (LSTM) recurrent neural network [8], and 2D CNN [23,20], but they are not designed for brain graph analysis. Here we choose to compare with BrainNetCNN [23], which is designed for fMRI network analysis. We also compare with the other GNN methods: [27], GraphSAGE [18], and our preliminary version PR-GNN. It is worth noting that GraphSAGE does not take edge weights in the aggregation step of the graph convolutional operation. The inputs of BrainNetCNN are correlation matrices. We follow the parameter settings indicated in the original paper [23]. The inputs and the settings of hidden layer nodes for the graph convolution, pooling and MLP layers of the alternative GNN methods are the same as BrainGNN. To answer Q5, we show the number of trainable parameters required by each method. Hyperparameters for baseline methods are tuned on validation datasets and we report the results on the 5 held-out testing datasets in Table 1.

As shown in Table 1, we report the comparison results using four different evaluation metrics, including accuracy, F1-score, recall and precision. On HCP dataset, the performance of our BrainGNN significantly suppresses the alternative methods ($p < 0.001$ under one tail two-sample t-test). On Biopoint dataset, BrainGNN is the significantly better than most of the alternative methods ($p < 0.05$ under one tail two-sample t-test) except for the previous version of our own work, PR-GNN and BrainGNN ($p < 0.3$ under one tail two-sample t-test). Although comparing to PR-GNN and BrainGNN, the mean values of different metrics are consistently better. The improvement may result from two causes. First, due to the intrinsic complexity of fMRI, complex models with more parameters are desired, which also explains why CNN and GNN-based methods were better than SVM and random forest. Second, our model utilized the properties of fMRI and community structure in the brain network and thus potentially modeled the local integration more effectively. Compared to alternative machine learning models, BrainGNN achieved significantly better classification results on two independent task-fMRI datasets. Moreover, BrainGNN does not have the burden of feature selection, which is needed in traditional machine learning methods. Compared with FNN and CNN-based methods, GNN-based methods require less trainable parameters. Specifically, BrainGNN needs only $10 - 30\%$ of the number of parameters compared to MLP and less than 3% of the number of parameters compared to BrainNetCNN. Hence, requiring less parameters and achieving higher data utility, Because BrainGNN is less parsimonious, it is more suitable as a deep learning tool for fMRI analysis, when the sample size is often limited.

3.5 Interpretability of BrainGNN

A compelling advantage of BrainGNN is its ***in-build*** interpretability. In this section, we would like to answer the following question:

Q6 What can be interpreted from BrainGNN? How can interpretability be used to gain insights into the data?

A brief answer to Q6 is that, (1) on one hand, users can interpret salient brain regions that are informative to the prediction task at different levels; (2) on the other hand, BrainGNN clusters brain regions into prediction-related communities. We demonstrate perspective (1) in Section 3.5- 3.5 and perspective (2) in Section 3.5. We show how our method can provide insights on the relation to finding salient ROIs, which can be treated as disease-related biomarkers or fingerprinting of cognitive states.

Individual- or Group- Level Biomarker? It is essential for a pipeline being able to discover personal biomarkers and group-level biomarkers in different application scenarios, i.e. precision medicine and disease understanding. In this section, we discuss how to adjust λ_2, the coefficient associated with GLC loss, to manipulate the level of biomarker interpretation through training.

Our proposed R-pool can prune the uninformative nodes and their connections from the brain graph based on the learning tasks. In other words, only the salient nodes are kept/selected. We investigate how to control the similarity between the selected ROIs of different individuals by tuning the coefficient λ_2. As we discuss in Section 2.4, large λ_2 encourages group-level interpretation (similar biomarkers across subjects) and small λ_2 encourages individual-level interpretation (various biomarkers across subjects). But when λ_2 is too large, the regularization might hurt the model accuracy (shown in Fig.4). We put forth the hypothesis that reliable interpretation can only be guaranteed in terms of a model with high classification accuracy. Hence, the interpretation is restricted to models with fixed $\lambda_1 = 0.1$ and varying λ_2 from 0 to 0.5 based on our experiments in Section 3.3. Without losing the generalizability, we show the salient ROI detection results of 3 randomly selected ASD instances from the Biopoint dataset in Fig. 5. We show the remaining 21 ROIs after the 2nd R-pool layer (with pooling ratio = 0.5, 25% nodes left) and corresponding pooling scores. As shown in Fig. 5(a), when $\lambda_2 = 0$, overlapped areas among the three instances are rarely to be found. In Fig. 5(b-c), we circle the big overlapped areas across the three instances. By visually examining the salient ROIs, we find three overlapped areas in Fig. 5(b) and five overlapped areas in Fig. 5(c).

Validating Salient ROIs To demonstrate the effectiveness of the interpreted salient ROIs, we compare the biomarker with existing literature studies. We average the node pooling scores after the 1st R-pool layer for all subjects per class and select the top 21 salient ROIs as biomarkers for that class.

In Fig. 6, we display the salient ROIs associated with HC and ASD separately. Putamen, thalamus, temporal gyrus and insular, occipital lobe are selected for HC; frontal gyrus, temporal lobe, cingulate gyrus, occipital pole, and angular gyrus are selected for ASD. Hippocampus and temporal pole are important for both groups. We name the selected ROIs as the biomarkers for identifying each group.

The biomarkers for HC corresponded to the areas of clear deficit in ASD, such as social communication, perception, and execution. In contrast, the biomark-

ers of ASD map to implicated activation-exhibited areas in ASD: default mode network [4] and memory [3]. This conclusion is consistent both with behavioral observations when administering the fMRI paradigm and with a prevailing theory that ASD includes areas of cognitive strengths amidst the social deficits [35,39,19].

In Fig. 7(a-g), we list the salient ROIs associated with the seven tasks for the HCP dataset. To validate the neurological significance of the result, we used Neurosynth [50], a platform for fMRI data analysis. Neurosynth collects thousands of neuroscience publications and provides meta-analysis, finding the keywords and their associated statistical images. The decoding function on the platform calculates the correlation between the input image and each functional keyword's meta-analysis images. A high correlation indicates large association between the salient ROIs and the functional keyword. We selected 'gambling', 'language', 'motor', 'relational', 'social', 'working memory' (WM) and 'emotion' as the functional keywords, which were exactly the functional design of the 7 tasks. The heatmap in Fig. 8 illustrates the meta-analysis on functional keywords implied by the top salient regions corresponding to the seven tasks using Neurosynth. We define a state set, which is the same as the functional keywords set, as $\mathcal{S}$ ={'language', 'motor', 'relational', 'social', 'WM' and 'emotion'}. In practice, given the interpreted salient ROIs associated with a functional state $fs \in \mathcal{S}$, we generate the corresponding binary ROI mask. The mask is used as the input for Neurosynth analysis, which generate a vector of association scores between salient ROIs of fs and all the keywords in $\mathcal{S}$ = as it is represented as a row in Fig. 8. To assist visualization, we normalize the values by dividing the maximum absolute value of each row. If the diagonal value (from bottom left to top right) is 1, it indicates the interpreted salient ROIs reflect its real task state. The finding in Fig. 8 suggests that our algorithm can identify ROIs that are key to distinguish between the 7 tasks. For example, the anterior temporal lobe and temporal parietal regions are selected for the social task, which are typically associated with social cognition [30,36].

Node Clustering Patterns in Ra-GConv layer From the best fold of each dataset, we cluster all the ROIs based on the kernel parameter α_{ij}^{+} (learned in Eq. (3)) of the 1st Ra-GConv layer, which indicates the membership score of the region i for community j. We show the node clustering results for Biopoint and HCP data in Fig. (9a) and Fig. (9b) respectively. Fig. 10 shows that the membership scores ($[\alpha_{ij}^{+}]$ matrices) are not uniformly distributed across the communities and only one or a few communities have significantly higher scores than the other communities for a given ROI. This corroborates the necessity of using different kernels to learn node representation by forming different communities. Also, $[\alpha_{ij}^{+}]$ can be treated as a multiplier, and a small value means suppression. We notice that the $[\alpha_{ij}^{+}]$ matrices are overall sparse. Only a few ROIs have high α_{ij}^{+} values. This is in good agreement with our hypothesis that R-pool tends to select a few representative ROIs to summarize the group-level representation.

4 Discussion

Our proposed BrainGNN includes novel Ra-GConv layers that efficiently assign each ROI a unique kernel, revealing ROI community patterns, and novel regulation terms (unit loss, GLC loss and TPK loss) for pooling operations that regulate the model to select salient ROIs. It shows superior prediction accuracy for ASD classification and brain states decoding compared to the alternative machine learning, MLP, CNN and GNN methods. As it is shown in Fig 1, BrainGNN improves average accuracy by 3% to 20% for ASD classification on the Biopoint dataset and achieves an average accuracy of 93.4% on a seven-states classification task on the HCP dataset.

Despite the high accuracy achieved by deep learning models, a natural question that arises is if the decision making process in deep learning models can be interpretable. From the brain biomarker detection perspective, understanding salient ROIs associated with the prediction is an important approach to find the biomarkers, where the indicative ROIs could be candidate biomarkers. Here, we use model interpretability to address the issue of group-level and individual-level biomarker analysis. In contrast, without additional post-processing steps, the existing methods of fMRI analysis can only either perform individual-level or group-level functional biomarker detection. For example, general linear model (GLM), principal component analysis (PCA) and independent component analysis (ICA) are group-based analysis methods. Some deterministic models like connectome-based predictive modeling (CPM) [38,14] (a coarse model averaging edge strengths over entire subject for prediction) and other machine learning based methods provide individual-level analysis. However, the model flexibility for different-levels of biomarkers analysis might be required by different users. For precision medicine, individual-level biomarkers are desired for planning targeted treatment, whereas group-level biomarkers are essential for understanding the common characteristic patterns associated with the disease. To fill the gap between group-level and individual-level biomarker analysis, we introduce a tunable regularization term for our graph pooling function. By examining the pairs of inputs and intermediate outputs from the pooling layers, our method can switch freely between individual-level and group-level explanation under the control of the regularization term by end-to-end training. A large regularization parameter for group consistency encourages interpreting common biomarkers for all the instances, while a small regularization parameter allows different interpretations for different instances. However, the appropriate coefficients are study-specific and the suitable range can be explored using validation dataset.

5 Limitation and Future Work

The pre-processing procedure performein Section 3.1 is one possible way of obtaining graphs from fMRI data, as demonstrated in this work. One meaningful next step is to use more powerful local feature extractors to summarize ROI information. A joint end-to-end training procedure that dynamically extracts

graph node features from fMRI data is challenging, but an interesting direction. Also, in the current work, we only try a single atlas for each dataset. In brain analysis, the reproducibility and consistency of the methods are important [46,1]. For ROI-based analysis, usually different atlases lead to different results [7]. It is worth further investigating whether the classification and interpretation results are robust to the choice of the atlas. Although we discussed a few variations of hyperparameters in Section 3.3, more variations should be studied, such as pooling ratio, the number of communities, the number of convolutional layers, and different readout operations. In future work, we will explore the connections between the Ra-GConv layer and the tensor decomposition-based clustering methods and the patterns of ROI selection and ROI clustering. For better understanding the algorithm, we aim to work on quantitative evaluations and theoretical studies to explain the experimental results.

6 Conclusions

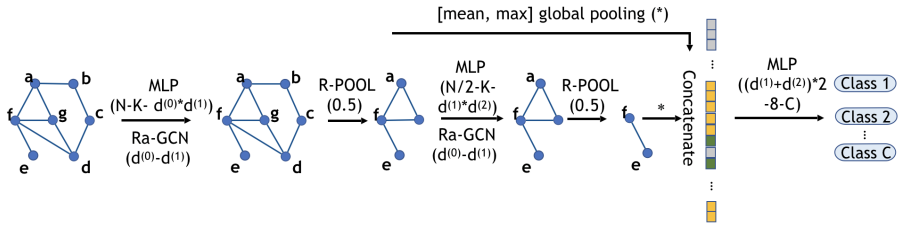
In this paper, we propose BrainGNN, an interpretable graph neural network for fMRI analysis. BrainGNN takes graphs built from neuroimages as inputs, and then outputs prediction results together with interpretation results. We applied BrainGNN on Biopoint and HCP fMRI datasets. With the built-in interpretability, BrainGNN not only performs better on prediction than alternative methods, but also detects salient brain regions associated with predictions and discovers brain community patterns. Overall, our model shows superiority over alternative graph learning and machine learning classification models. By investigating the selected ROIs after R-pool layers, our study reveals the salient ROIs to identify autistic disorders from healthy controls and decodes the salient ROIs associated with certain task stimuli. Certainly, our framework is generalizable to analysis of other neuroimaging modalities. The advantages are essential for developing precision medicine, understanding neurological disorders, and ultimately benefiting neuroimaging research.

7 Declaration of Competing Interest

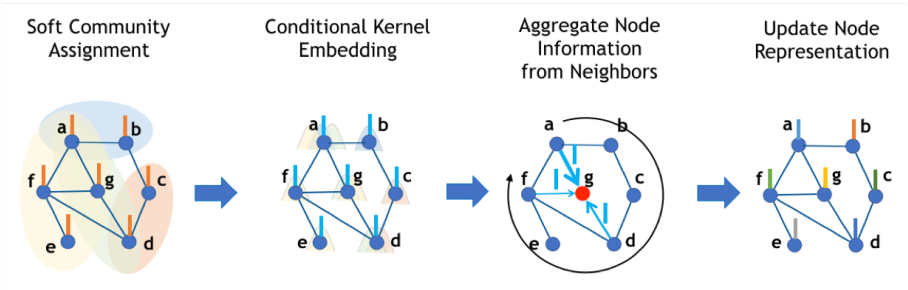
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

8 Acknowledgements

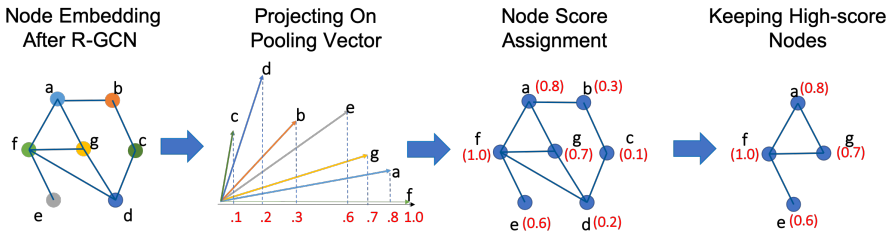
Parts of this research was supported by National Institutes of Health (NIH) [R01NS035193, R01MH100028].



(a) BrainGNN Architecture



(b) Operations in Ra-GConv layer



(c) Operations in R-pool Layer

Fig. 2: (a) introduces the BrainGNN architecture that we propose in this work. BrainGNN is composed of blocks of Ra-GConv layers and R-pool layers. It takes graphs as inputs and outputs graph-level predictions. (b) shows how the Ra-GConv layer embeds node features. First, nodes are softly assigned to communities based on their membership scores to the communities. Each community is associated with a different basis vector. Each node is embedded by the particular basis vectors based on the communities that it belongs to. Then, by aggregating a node's own embedding and its neighbors' embedding, the updated representation is assigned to each node on the graph. (c) shows how R-pool selects nodes to keep. First, all the nodes' representations are projected to a learnable vector. The nodes with large projected values are retained with their corresponding connections.

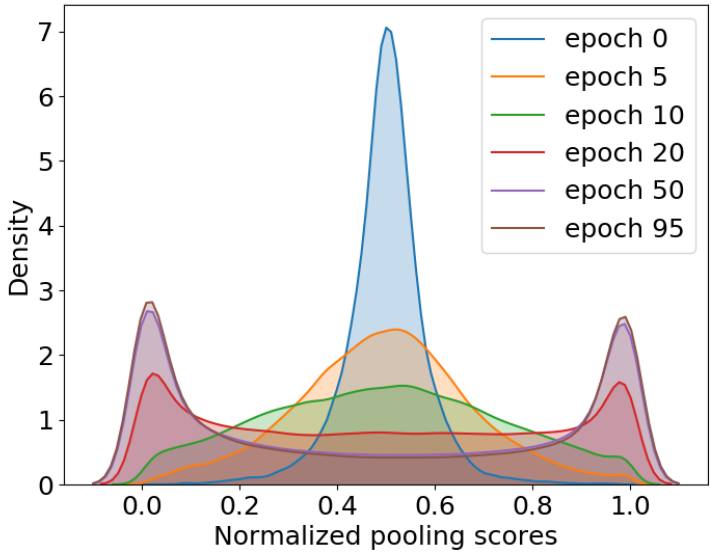
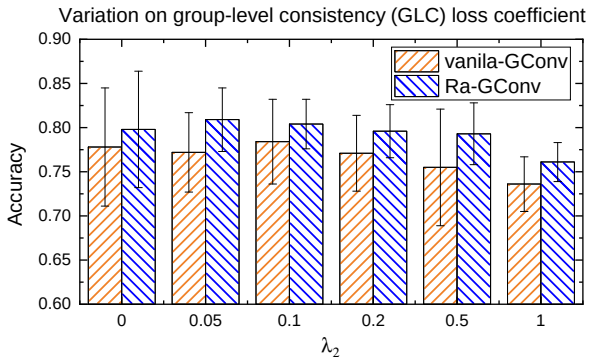
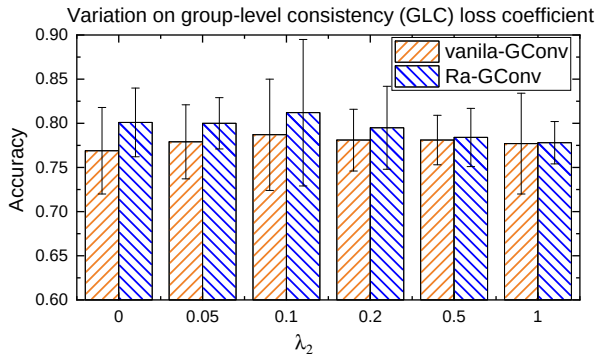


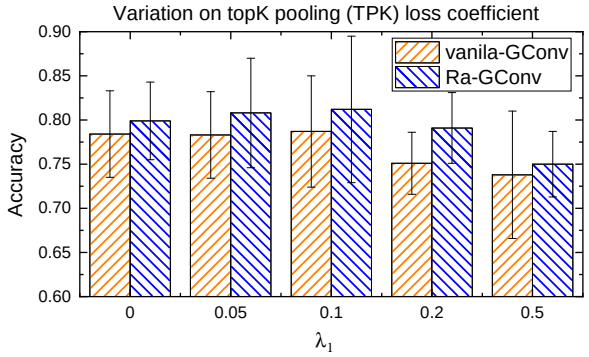
Fig. 3: The change of the distribution of node pooling scores $\hat{\mathbf{s}}$ of the 1st R-pool layer over 100 training epochs. With TopK pooling (TPK) loss, the node pooling scores of the selected nodes and those of the unselected nodes become significantly separate.



(a) Variation on λ_2, when setting $\lambda_1 = 0$



(b) Variation on λ_2, when setting $\lambda_1 = 0.1$



(c) Variation on λ_1, when setting $\lambda_2 = 0.1$

Fig. 4: Compare Ra-GConv with vanilla-GConv and discuss coefficients of total loss in terms of accuracies for 5-fold study.

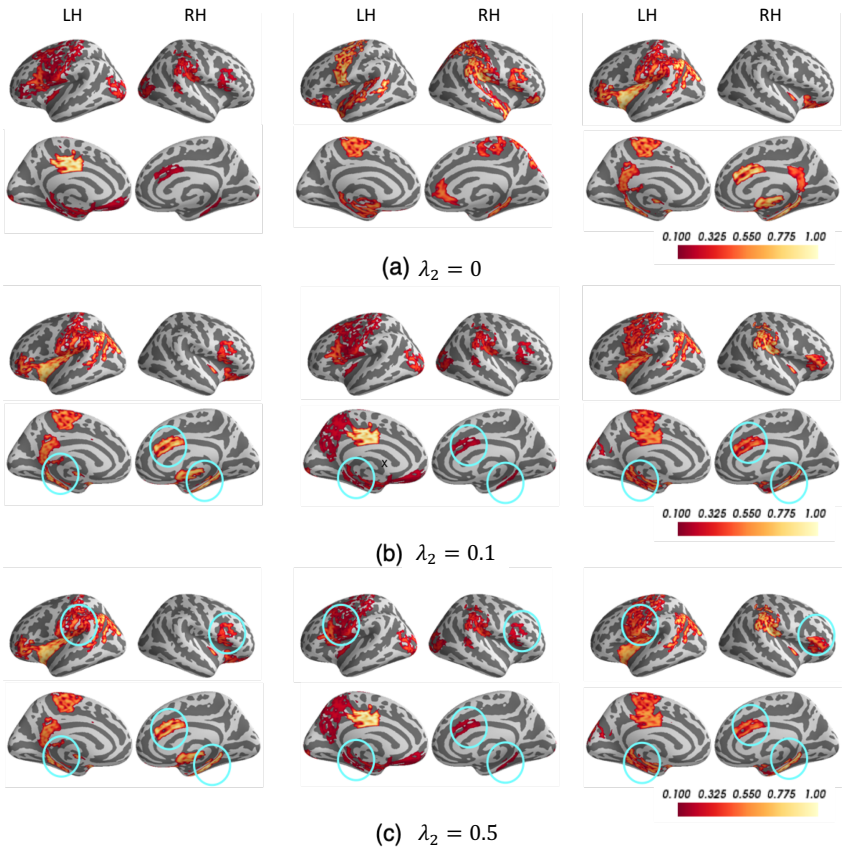
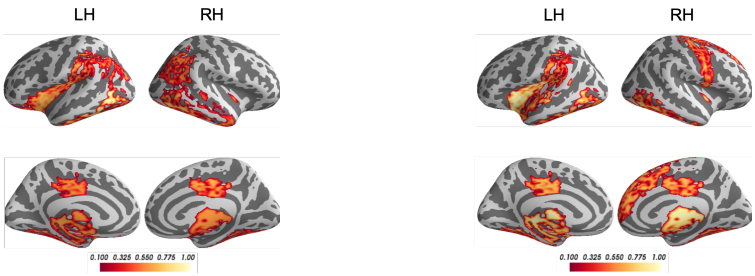
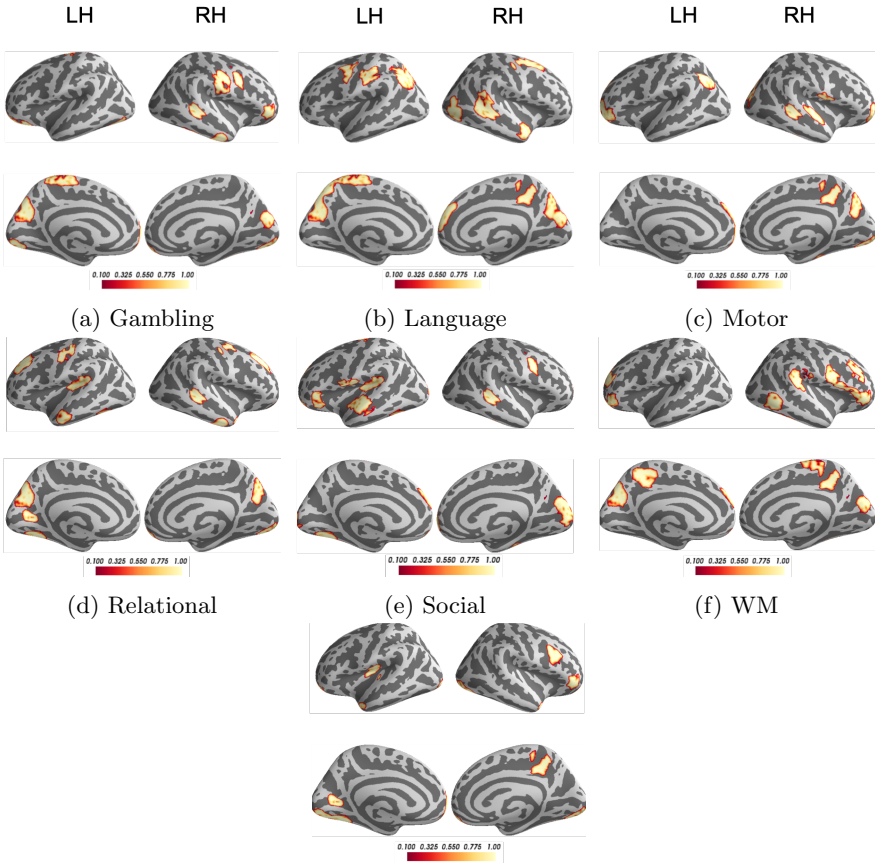


Fig. 5: The selected salient ROIs of three different ASD individuals with different weights λ_2 associated with group-level consistency term L_{GLC}. The color bar ranges from 0.1 to 1. The bright-yellow color indicates a high score, while dark-red color indicates a low score. The common detected salient ROIs across different individuals are circled in blue.



(a) HC (b) ASD

Fig. 6: Interpreting salient ROIs (importance scores are denoted in colorbar) for classifying HC vs. ASD using BrainGNN.



(a) Gambling (b) Language (c) Motor

(d) Relational (e) Social (f) WM

(g) Emotion

Fig. 7: Interpreting salient ROIs (importance scores are denoted in color-bar) associated with classifying seven tasks.

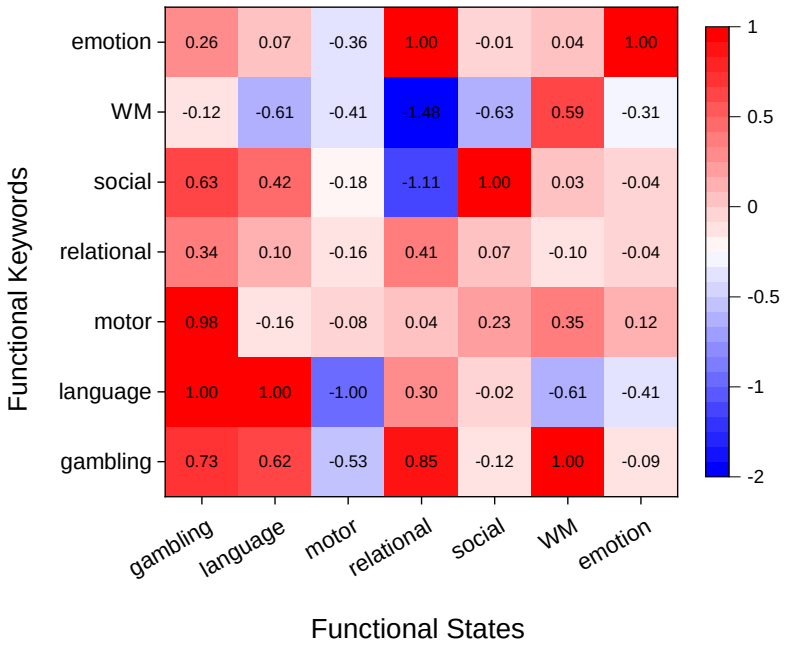
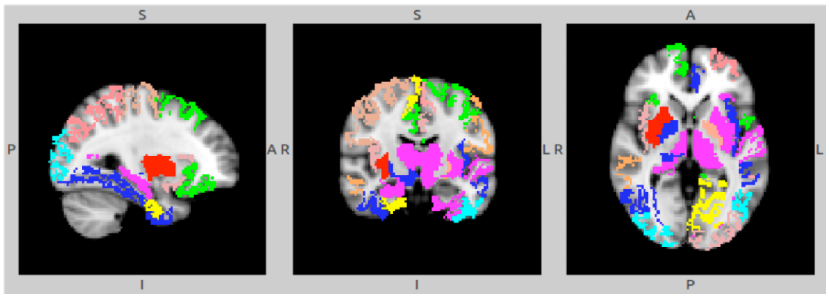
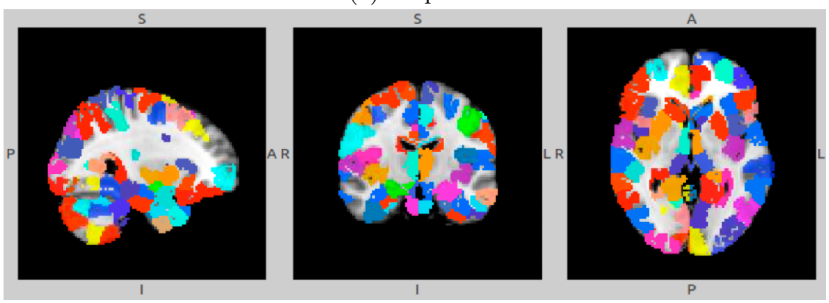


Fig. 8: The correlation scores (normalized by dividing largest absolute value of each row) the interpreted biomarkers associated with the functional keywords using NeuroSynth for each functional state. A high correlation indicates large association between the salient ROIs and the functional keyword.



(a) Biopoint



(b) HCP

Fig. 9: ROI clustering learned from α_{ij}^{+} from the 1st Ra-GConv layer. Different colors denote different communities.

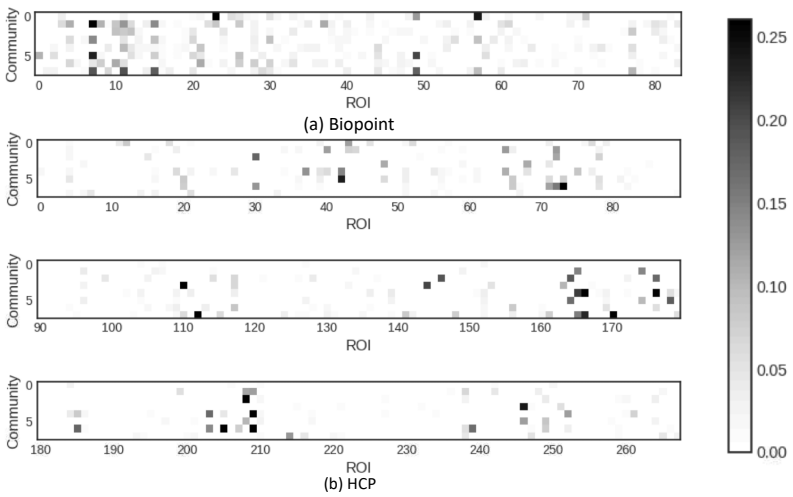


Fig. 10: Visualizing Ra-GConv parameter α^{+}, which implies the membership score of a ROI to a community.

References

1. Abraham, A., Milham, M.P., Di Martino, A., Craddock, R.C., Samaras, D., Thirion, B., Varoquaux, G.: Deriving reproducible biomarkers from multi-site resting-state data: an autism-based example. NeuroImage **147**, 736–745 (2017)
2. Baker, J.T., Holmes, A.J., Masters, G.A., Yeo, B.T., Krienen, F., Buckner, R.L., Öngür, D.: Disruption of cortical association networks in schizophrenia and psychotic bipolar disorder. JAMA psychiatry **71**(2), 109–118 (2014)
3. Boucher, J., Bowler, D.M.: Memory in autism. Citeseer (2008)
4. Buckner, R.L., Andrews-Hanna, J.R., Schacter, D.L.: The brain's default network: anatomy, function, and relevance to disease. (2008)
5. Cai, C., Wang, Y.: A note on over-smoothing for graph neural networks. arXiv preprint arXiv:2006.13318 (2020)
6. Cangea, C., et al.: Towards sparse hierarchical graph classifiers. arXiv preprint arXiv:1811.01287 (2018)
7. Dadi, K., Rahim, M., Abraham, A., Chyzhyk, D., Milham, M., Thirion, B., Varoquaux, G., Initiative, A.D.N., et al.: Benchmarking functional connectome-based predictive models for resting-state fmri. Neuroimage **192**, 115–134 (2019)
8. Dakka, J., Bashivan, P., Gheiratmand, M., Rish, I., Jha, S., Greiner, R.: Learning neural markers of schizophrenia disorder using recurrent neural networks. arXiv preprint arXiv:1712.00512 (2017)
9. Desikan, R.S., Ségonne, F., Fischl, B., Quinn, B.T., Dickerson, B.C., Blacker, D., Buckner, R.L., Dale, A.M., Maguire, R.P., Hyman, B.T., et al.: An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. Neuroimage **31**(3), 968–980 (2006)
10. Du, Y., Fu, Z., Calhoun, V.D.: Classification and prediction of brain disorders using functional connectivity: promising but challenging. Frontiers in neuroscience **12**, 525 (2018)
11. Finn, E.S., Shen, X., Scheinost, D., Rosenberg, M.D., Huang, J., Chun, M.M., Papademetris, X., Constable, R.T.: Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. Nature neuroscience **18**(11), 1664 (2015)
12. Gan, J., Zhu, X., Hu, R., Zhu, Y., Ma, J., Peng, Z., Wu, G.: Multi-graph fusion for functional neuroimaging biomarker detection. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. pp. 580–586. International Joint Conferences on Artificial Intelligence Organization (7 2020), main track
13. Gao, H., Ji, S.: Graph u-nets. arXiv preprint arXiv:1905.05178 (2019)
14. Gao, S., Greene, A.S., Constable, R.T., Scheinost, D.: Combining multiple connectomes improves predictive modeling of phenotypic measures. Neuroimage **201**, 116038 (2019)
15. Gong, L., Cheng, Q.: Exploiting edge features for graph neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9211–9219 (2019)
16. Gopinath, K., Desrosiers, C., Lombaert, H.: Adaptive graph convolution pooling for brain surface analysis. In: International Conference on Information Processing in Medical Imaging. pp. 86–98. Springer (2019)
17. Greene, A.S., Gao, S., Scheinost, D., Constable, R.T.: Task-induced brain state manipulation improves prediction of individual traits. Nature communications **9**(1), 1–13 (2018)

18. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in neural information processing systems. pp. 1024–1034 (2017)
19. Iuculano, T., Rosenberg-Lee, M., Supekar, K., Lynch, C.J., Khouzam, A., Phillips, J., Uddin, L.Q., Menon, V.: Brain organization underlying superior mathematical abilities in children with autism. Biological Psychiatry **75**(3), 223–230 (2014)
20. Jie, B., Liu, M., Lian, C., Shi, F., Shen, D.: Designing weighted correlation kernels in convolutional neural networks for functional connectivity based brain disease diagnosis. Medical Image Analysis p. 101709 (2020)
21. Kaiser, M.D., Hudac, C.M., Shultz, S., Lee, S.M., Cheung, C., Berken, A.M., Deen, B., Pitskel, N.B., Sugrue, D.R., Voos, A.C., et al.: Neural signatures of autism. Proceedings of the National Academy of Sciences **107**(49), 21223–21228 (2010)
22. Karwowski, W., Vasheghani Farahani, F., Lighthall, N.: Application of graph theory for identifying connectivity patterns in human brain networks: a systematic review. frontiers in Neuroscience **13**, 585 (2019)
23. Kawahara, J., Brown, C.J., Miller, S.P., Booth, B.G., Chau, V., Grunau, R.E., Zwicker, J.G., Hamarneh, G.: Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment. NeuroImage **146**, 1038–1049 (2017)
24. Kazi, A., Shekarforoush, S., Krishna, S.A., Burwinkel, H., Vivar, G., Kortüm, K., Ahmadi, S.A., Albarqouni, S., Navab, N.: Inceptiongcn: receptive field aware graph convolutional network for disease prediction. In: International Conference on Information Processing in Medical Imaging. pp. 73–85. Springer (2019)
25. Kim, B.H., Ye, J.C.: Understanding graph isomorphism network for brain mr functional connectivity analysis. arXiv preprint arXiv:2001.03690 (2020)
26. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
27. Li, X., Dvornek, N.C., Zhou, Y., Zhuang, J., Ventola, P., Duncan, J.S.: Graph neural network for interpreting task-fmri biomarkers. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 485–493. Springer (2019)
28. Li, X., Zhou, Y., Dvornek, N.C., Zhang, M., Zhuang, J., Ventola, P., Duncan, J.S.: Pooling regularized graph neural network for fmri biomarker analysis. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 625–635. Springer (2020)
29. Loe, C.W., Jensen, H.J.: Comparison of communities detection algorithms for multiplex. Physica A: Statistical Mechanics and its Applications **431**, 29–45 (2015)
30. Mar, R.A.: The neural bases of social cognition and story comprehension. Annual review of psychology **62**, 103–134 (2011)
31. McClure, P., Moraczewski, D., Lam, K.C., Thomas, A., Pereira, F.: Evaluating adversarial robustness for deep neural network interpretability using fmri decoding. arXiv preprint arXiv:2004.11114 (2020)
32. Moğultay, H., Alkan, S., Yarman-Vural, F.T.: Classification of fmri data by using clustering. In: 2015 23nd Signal Processing and Communications Applications Conference (SIU). pp. 2381–2383. IEEE (2015)
33. Nandakumar, N., Manzoor, K., Pillai, J.J., Gujar, S.K., Sair, H.I., Venkataraman, A.: A novel graph neural network to localize eloquent cortex in brain tumor patients from resting-state fmri connectivity. In: International Workshop on Connectomics in Neuroimaging. pp. 10–20. Springer (2019)
34. Oono, K., Suzuki, T.: Graph neural networks exponentially lose expressive power for node classification. arXiv preprint arXiv:1905.10947 (2019)

35. Robertson, C.E., Kravitz, D.J., Freyberg, J., Baron-Cohen, S., Baker, C.I.: Tunnel vision: sharper gradient of spatial attention in autism. Journal of Neuroscience **33**(16), 6776–6781 (2013)
36. Ross, L.A., Olson, I.R.: Social cognition and the anterior temporal lobes. Neuroimage **49**(4), 3452–3462 (2010)
37. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference. pp. 593–607. Springer (2018)
38. Shen, X., Finn, E.S., Scheinost, D., Rosenberg, M.D., Chun, M.M., Papademetris, X., Constable, R.T.: Using connectome-based predictive modeling to predict individual behavior from brain connectivity. nature protocols **12**(3), 506 (2017)
39. Turkeltaub, P.E., Flowers, D.L., Verbalis, A., Miranda, M., Gareau, L., Eden, G.F.: The neural basis of hyperlexic reading: An fmri case study. Neuron **41**(1), 11–25 (2004)
40. Van Essen, D.C., Smith, S.M., Barch, D.M., Behrens, T.E., Yacoub, E., Ugurbil, K., Consortium, W.M.H., et al.: The wu-minn human connectome project: an overview. Neuroimage **80**, 62–79 (2013)
41. Veličković, P., et al.: Graph attention networks. In: ICLR (2018)
42. Venkataraman, A., Yang, D.Y.J., Pelphrey, K.A., Duncan, J.S.: Bayesian community detection in the space of group-level functional differences. IEEE transactions on medical imaging **35**(8), 1866–1882 (2016)
43. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing **17**(4), 395–416 (2007)
44. Wang, J., Zuo, X., He, Y.: Graph-based network analysis of resting-state functional mri. Frontiers in systems neuroscience **4**, 16 (2010)
45. Wang, X., Liang, X., Jiang, Z., Nguchu, B.A., Zhou, Y., Wang, Y., Wang, H., Li, Y., Zhu, Y., Wu, F., et al.: Decoding and mapping task states of the human brain via deep learning. Human Brain Mapping (2019)
46. Wei, X., Warfield, S.K., Zou, K.H., Wu, Y., Li, X., Guimond, A., Mugler III, J.P., Benson, R.R., Wolfson, L., Weiner, H.L., et al.: Quantitative analysis of mri signal abnormalities of brain white matter with high reproducibility and accuracy. Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine **15**(2), 203–209 (2002)
47. 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. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 772–782 (2019)
48. Yang, H., Li, X., Wu, Y., Li, S., Lu, S., Duncan, J.S., Gee, J.C., Gu, S.: Interpretable multimodality embedding of cerebral cortex using attention graph network for identifying bipolar disorder. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 799–807. Springer (2019)
49. Yang, X., Jin, Y., Chen, X., Zhang, H., Li, G., Shen, D.: Functional connectivity network fusion with dynamic thresholding for mci diagnosis. In: International Workshop on Machine Learning in Medical Imaging. pp. 246–253. Springer (2016)
50. Yarkoni, T., Poldrack, R.A., Nichols, T.E., Van Essen, D.C., Wager, T.D.: Large-scale automated synthesis of human functional neuroimaging data. Nature methods **8**(8), 665 (2011)