General Artificial Intelligence (1)

SAIR 2-02 Transformer for fMRI and its Application to Mental Disease Prediction

Momiao Xiong

Society of Artificial Intelligence Research

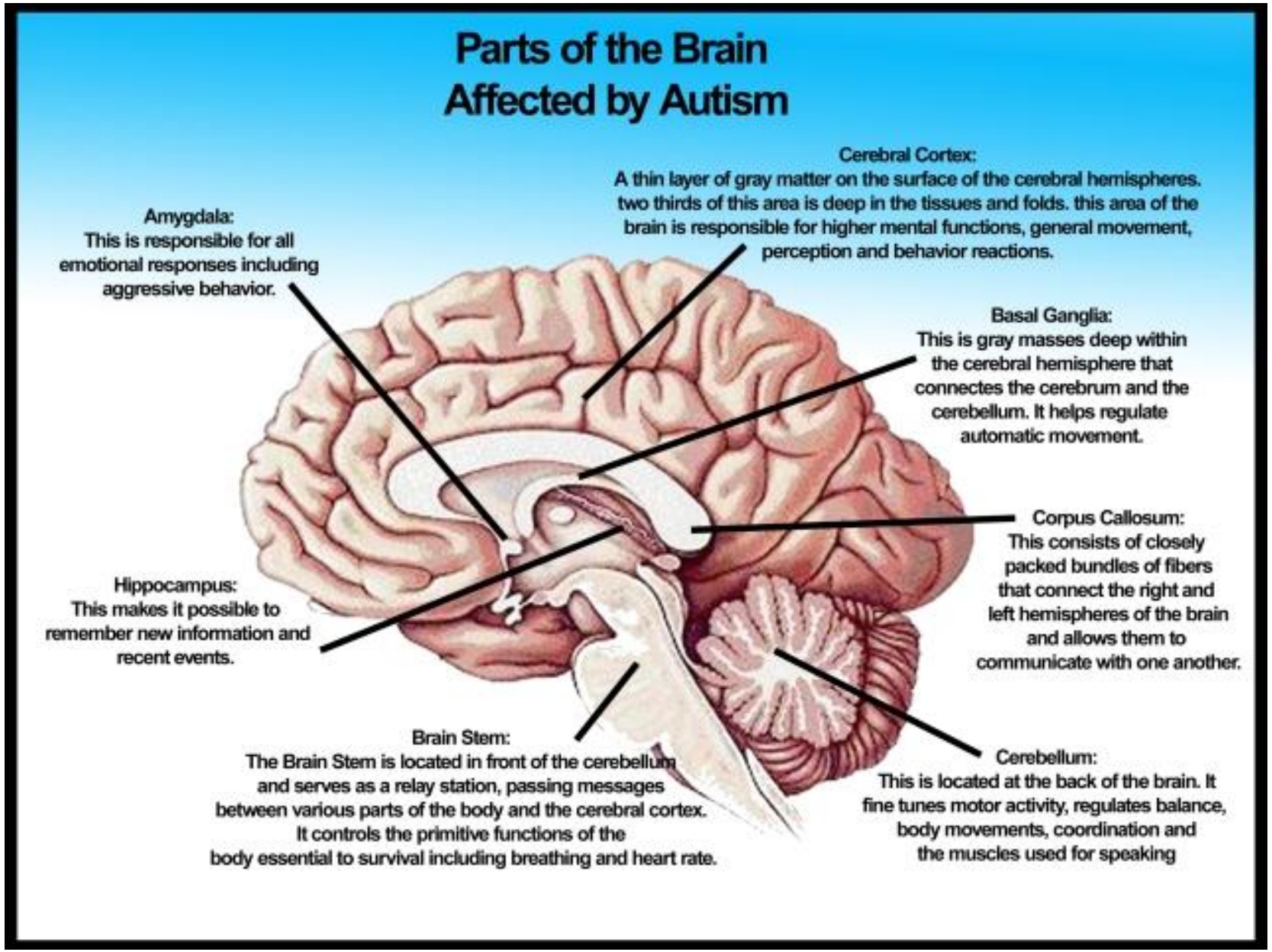
Community-Aware Transformer for Autism Prediction in fMRI Connectome

This lecture focuses on explaining the above paper

- **ASD diagnosis is challenging as there are no definitive medical tests such as a blood tests, and clinicians rely on behavioural and developmental assessments to accurately diagnose ASD**

- **As a more objective alternative measurement, neuroimaging tools, such as fMRI has been widely used to derive and validate biomarkers associated with ASD**

- **Conventionally, fMRI data is modelled as a functional connectivity (FC) matrix, e.g., by calculating Pearson correlation coefficients of pairwise brain ROIs**



Much like a computer, the brain relies on intricate wiring to process and transmit information. Scientists have discovered that in people with autism, this wiring is faulty, leading to misfiring in communications between brain cells.

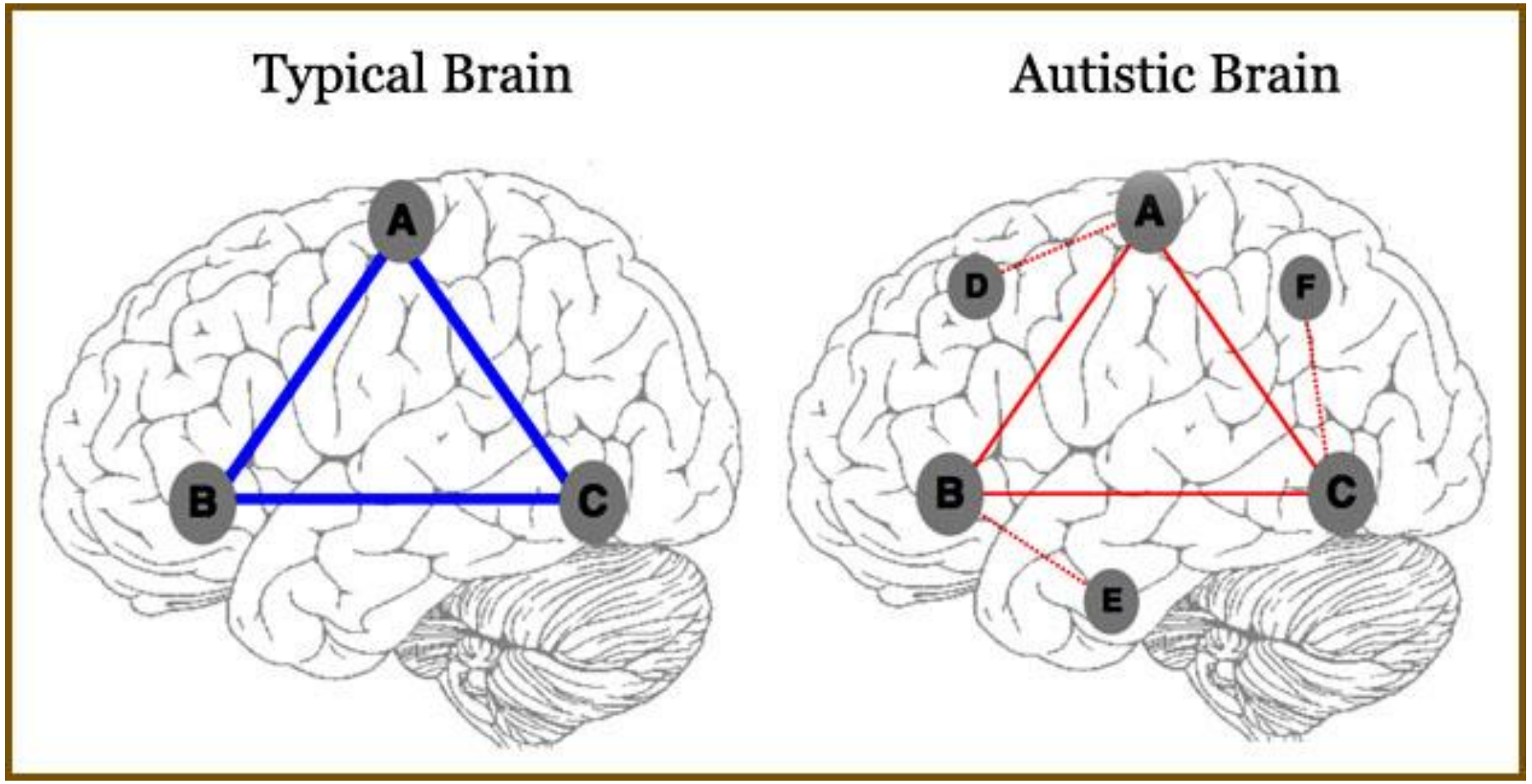
Abnormal brain areas in people with autism include the:

Cerebellum – reduced size in parts of the cerebellum.

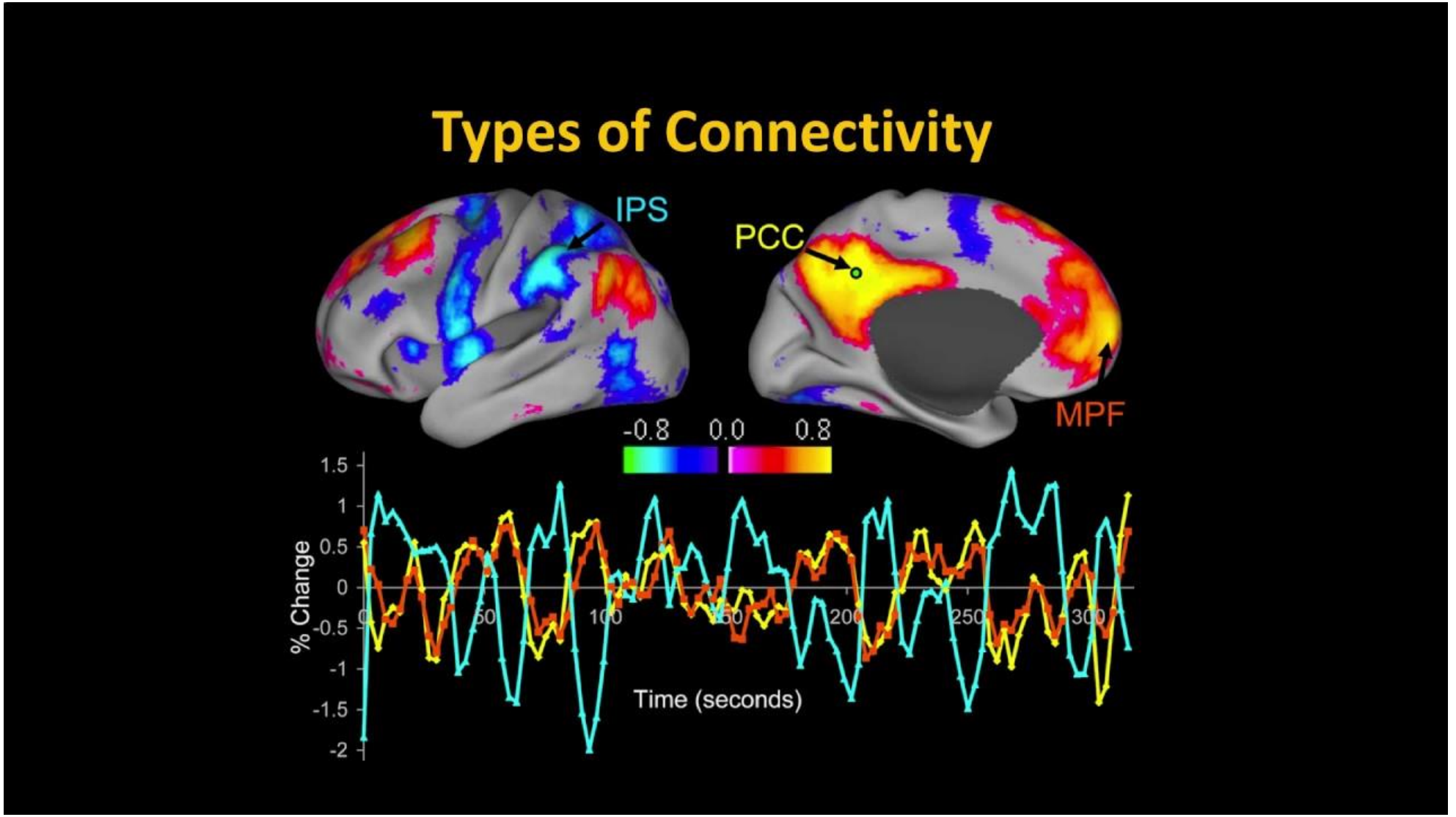
Hippocampus and **Amygdala** – smaller volume. Also, neurons in these areas are smaller and more tightly packed (higher cell density).

Lobes of the Cerebrum – larger size than normal. Ventricles – increased size.

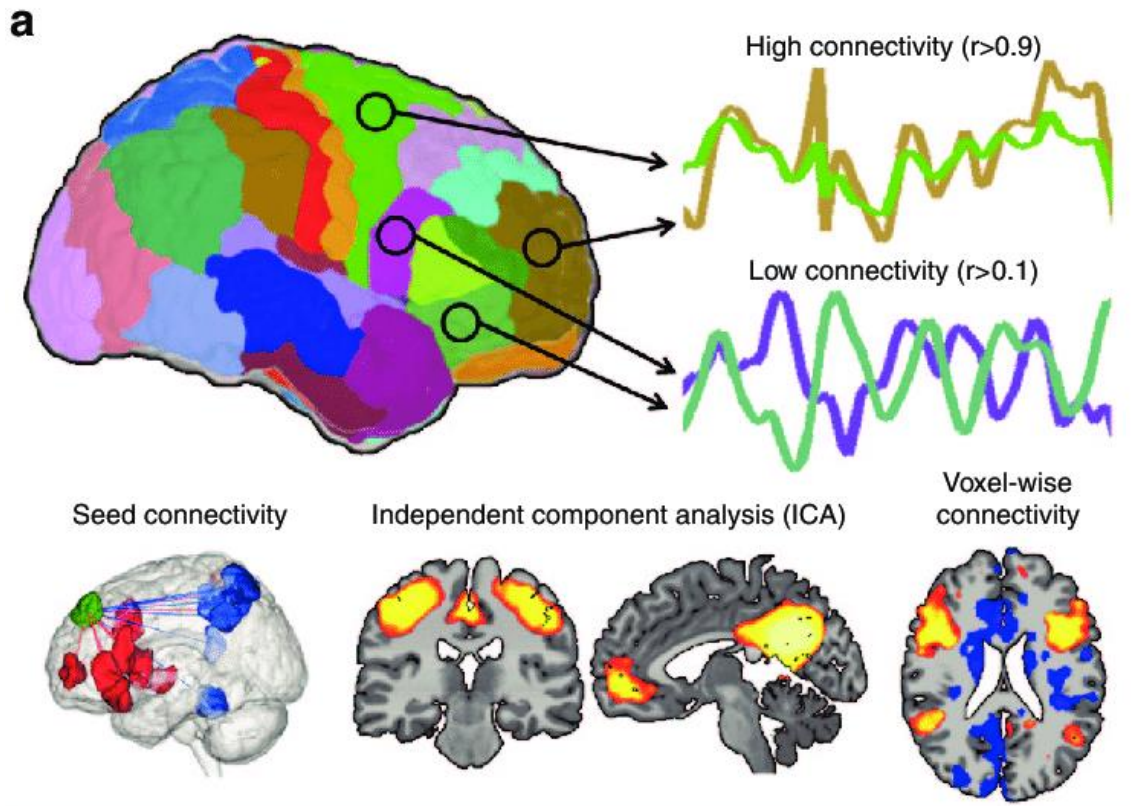
Caudate nucleus – reduced volume.

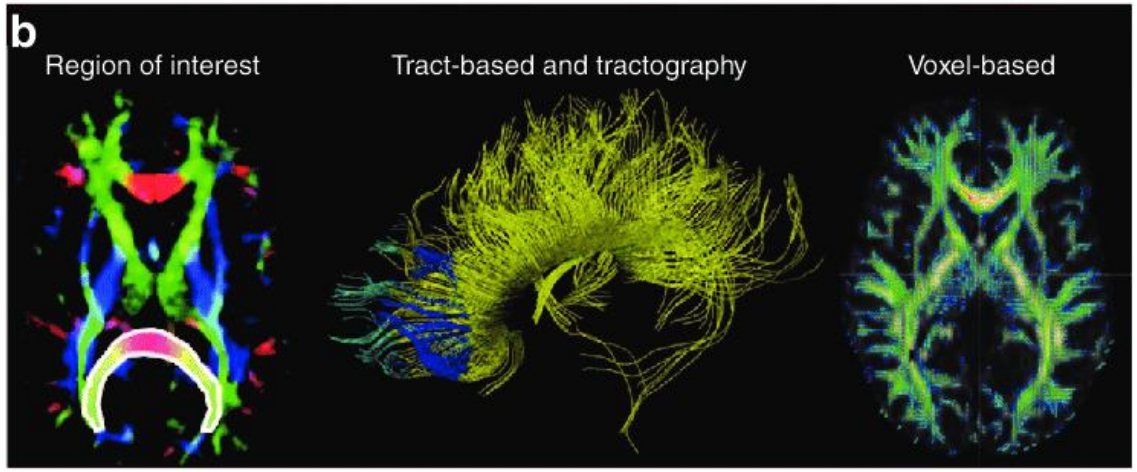


Overconnectivity in Brain Found in Autism

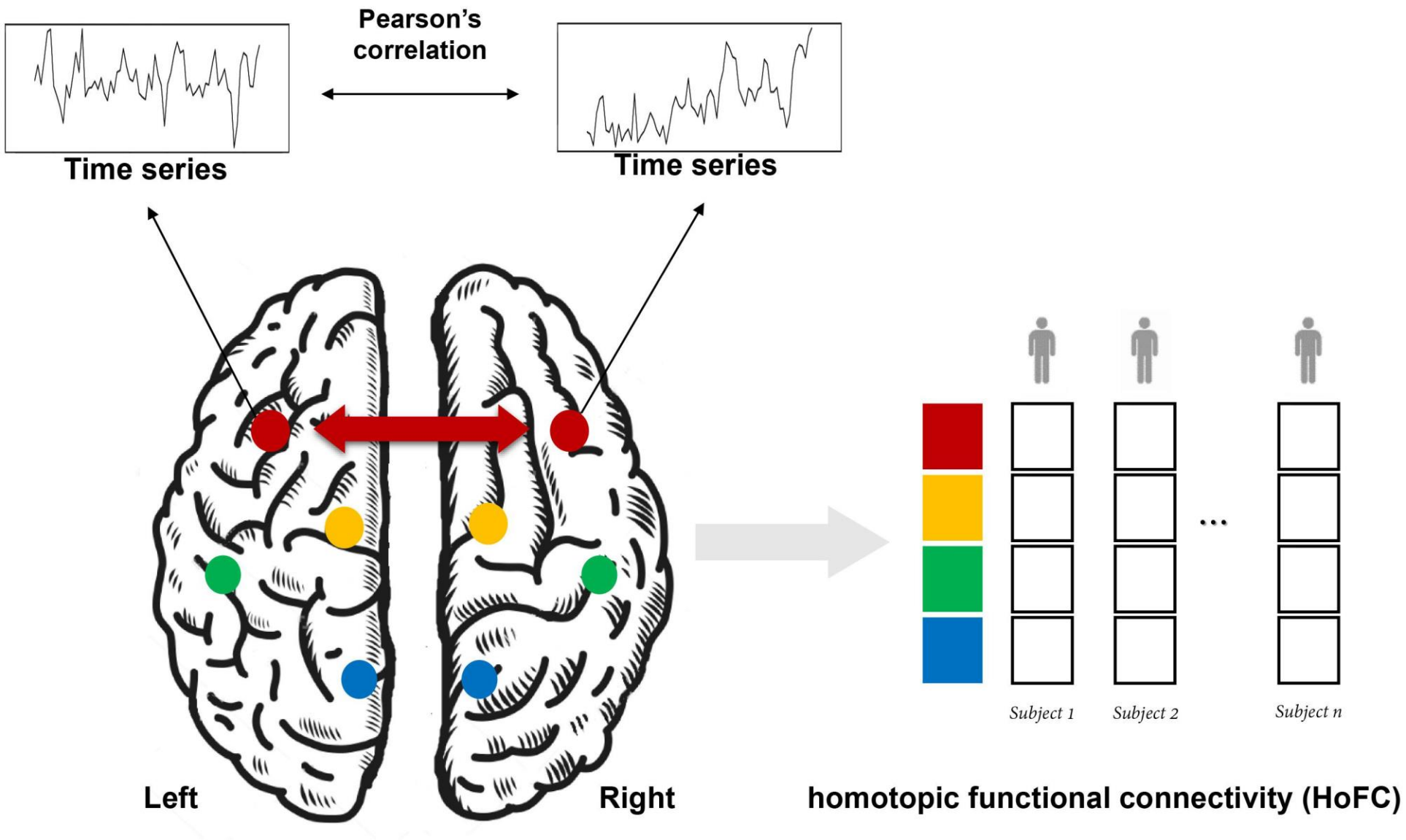


[UMass Amherst CONN Workshop, Part 1: Functional Connectivity]



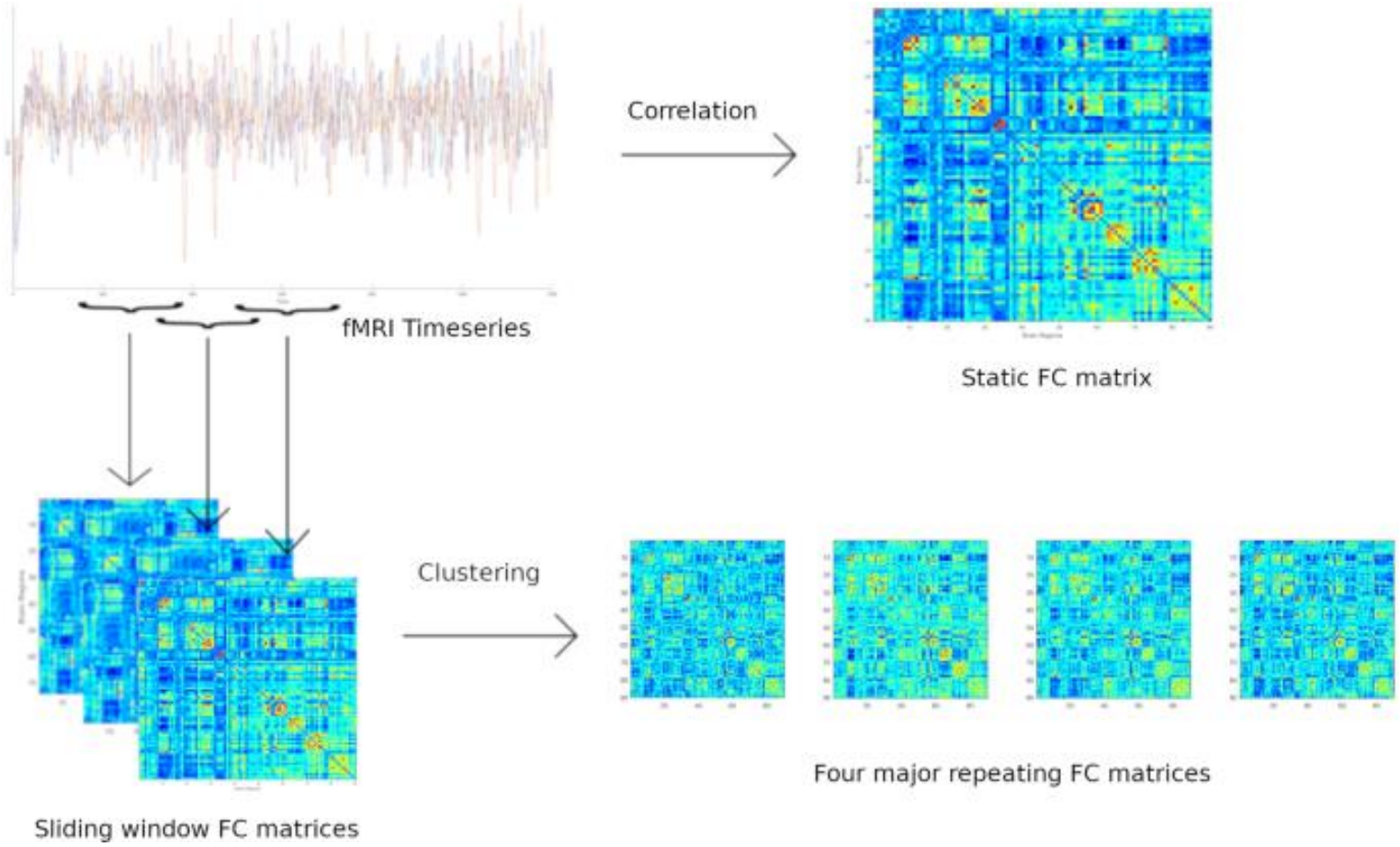


Examples of functional and structural connectivity. (a) Functional…



Functional Integration Between the Two Brain Hemispheres: Evidence From the Homotopic Functional Connectivity Under Resting State

functional connectivity (FC)



Static and dynamic FC matrices derived from fMRI time series. Static FC was calculated using Pearson correlation coefficients of the entire time series; however, dynamic FC was calculated considering a moving window of the time series and finding the major repeating FC matrices using a clustering algorithm.

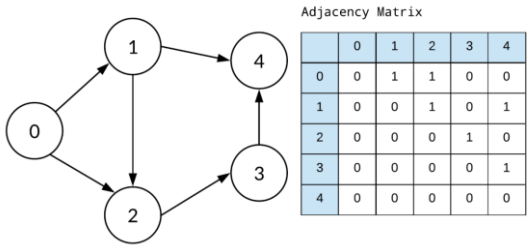
Deep learning (DL) models for fMRI-based brain connectome analysis

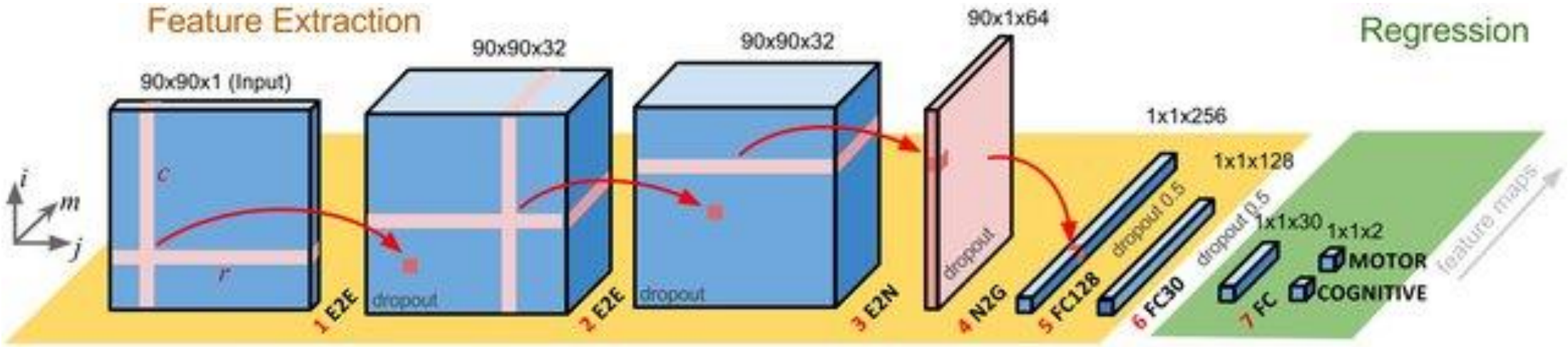
- **BrainNetCNN**

Edge-to-Edge (E2E) layer

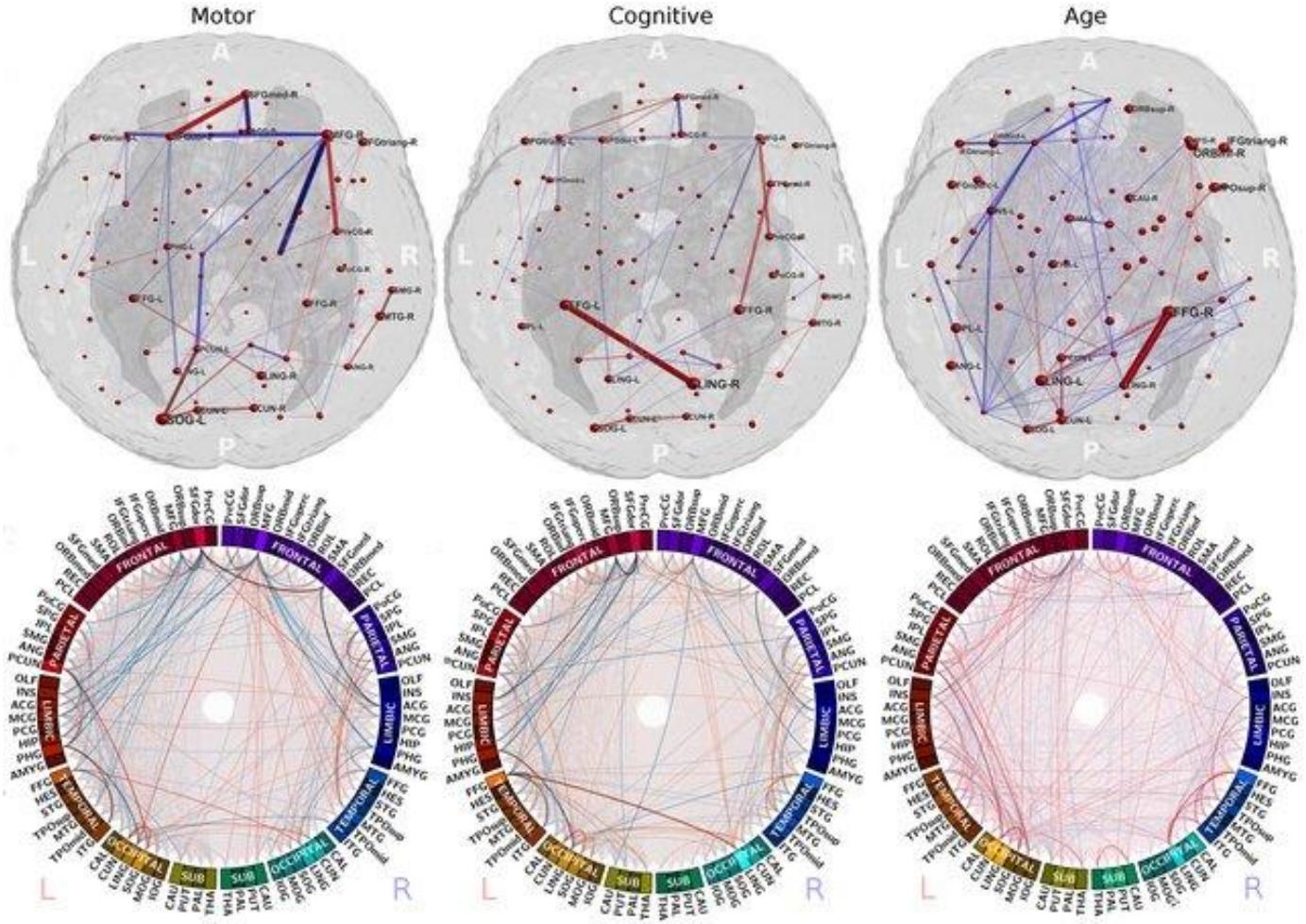
Edge-to-Node (E2N

Node-to-Graph (N2G) layer





Schematic representation of the BrainNetCNN architecture. Each block represents the input and/or output of the numbered filter layers. The 3rd dimension of each block (i.e., along vector m) represents the number of feature maps, M , at that stage. The brain network adjacency matrix (leftmost block) is first convolved with one or more (two in this case) E2E filters which weight edges of adjacent brain regions. The response is convolved with an E2N filter which assigns each brain region a weighted sum of its edges. The N2G assigns a single response based on all the weighted nodes. Finally, fully connected (FC) layers reduce the number of features down to two output score predictions.

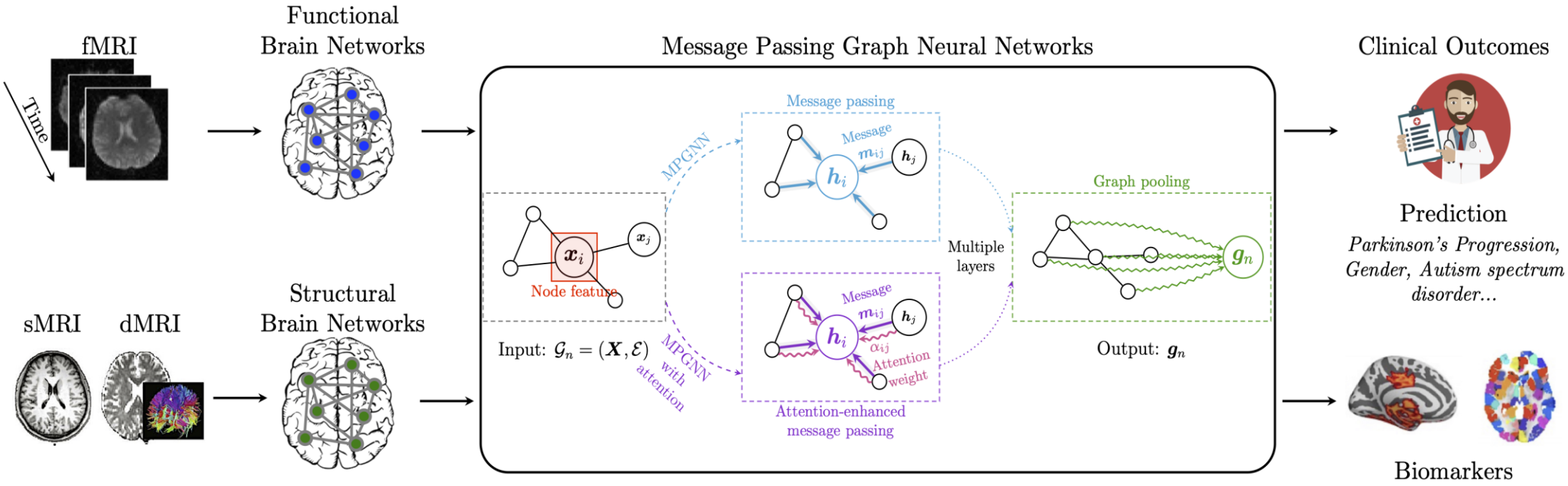


$$\frac{\partial \text{outcome}}{\partial w_{ij}}$$

Connections learned by BrainNetCNN to be most predictive of outcomes and ages. Top Row: **Edges with positive (red) and negative (blue) partial derivatives** with respect to motor outcomes (left), cognitive outcomes (middle) and ages (right). Edge **thickness and opacity** represent the **magnitude of each partial derivative**. Very small magnitudes (< 0.001) were omitted for clarity. **Node sizes** represent **the sum of partial derivative** magnitudes of **all neighbouring edges** with positive derivatives. Bottom row: The same partial derivatives mapped on to Circos ideograms. **Brightness of the color of the regions** in each ring denotes the **sum of positive partial derivative** magnitudes.

Brain Graph Neural Networks (BGNN)

- graph neural networks (GNNs) have emerged as a promising method for analyzing and modelling brain connectome by constructing graph representation from FC matrices

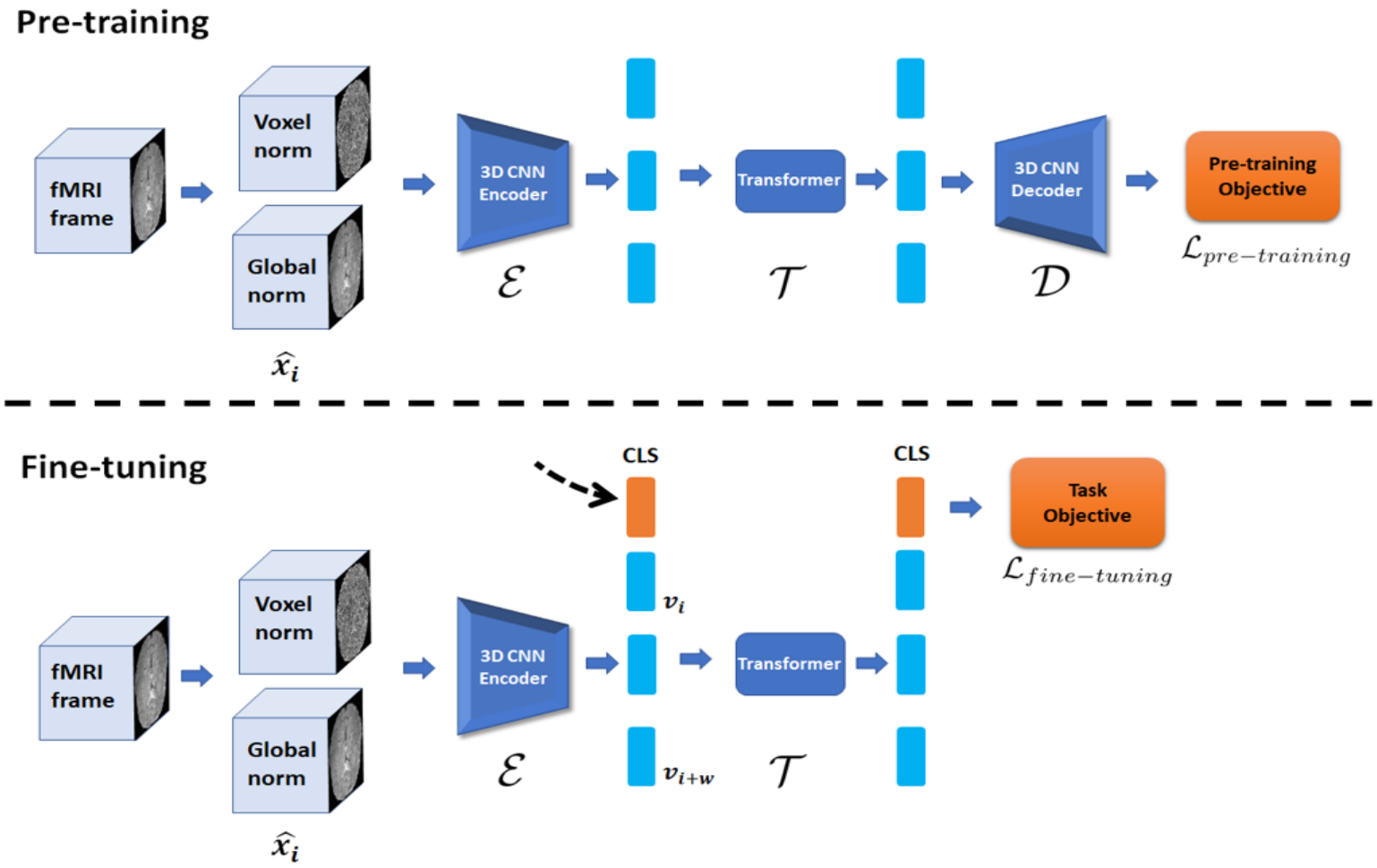


Cui H et al. 2023. Brain Network Analysis with Graph Neural Network

the source code o BrainGB at https://github.com/HennyJie/BrainGB

https://braingb.us/get-started/

Pre-training Network Transformer for fMRI

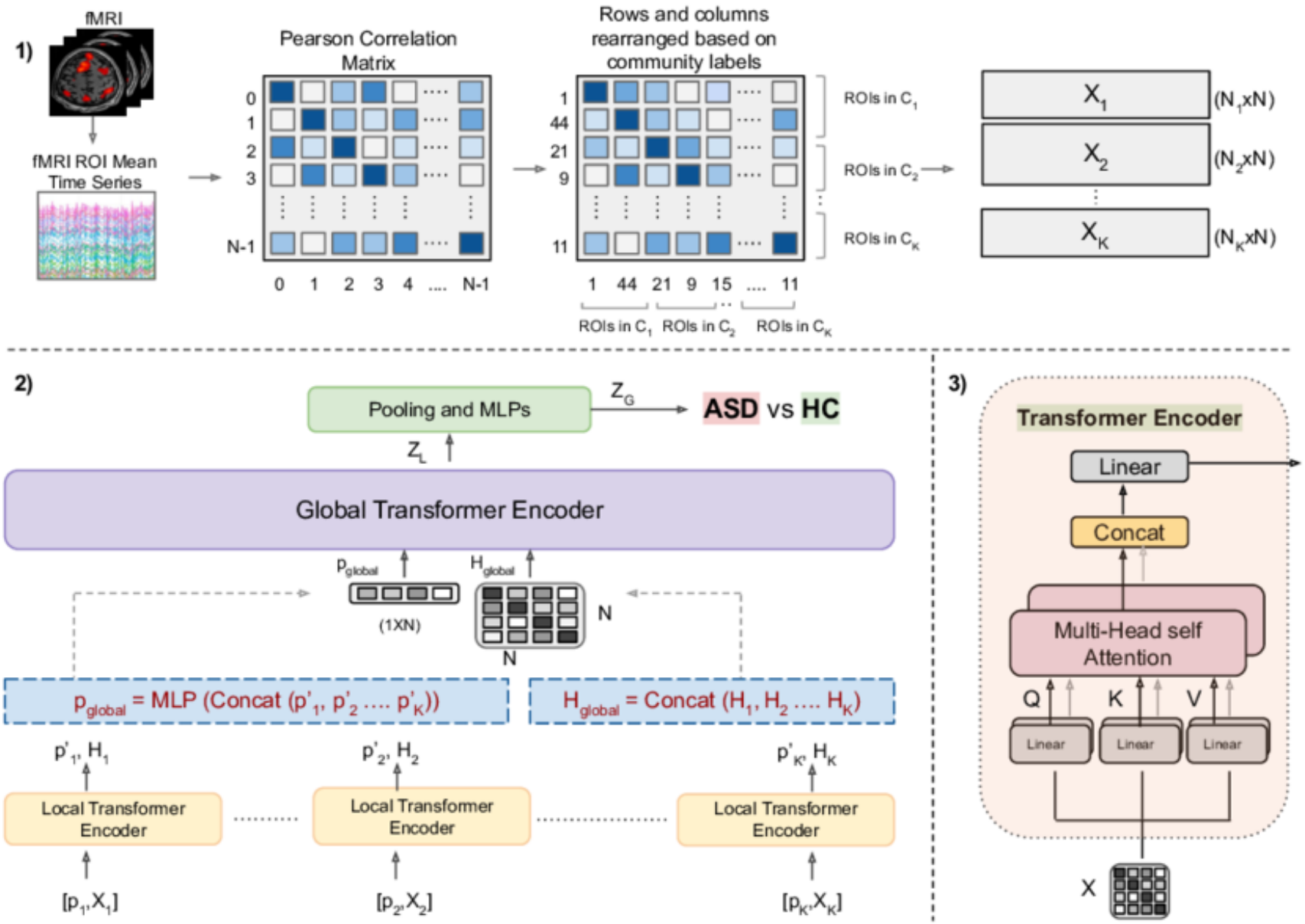


During pre-training (top), a sequence of fMRI frames are normalized and propagate through the 3D CNN encoder network . The encoder outputs a single vector for each input frame. The vectors are grouped into a unified sequence and propagated through the transformer network The output sequence is decomposed, aggregated on the batch dimension, and propagated through the decoder, which reconstructs the input data. During fine-tuning (bottom), a sequence of fMRI frames are propagated separately through the pre-trained encoder network. The vectors are aggregated as a sequence and a special CLS token is concatenated to the beginning of the sequence. The entire sequence is then propagated through the transformer model, which utilizes the embedding of the CLS token for making predictions.

Pre-training and Fine-tuning Transformers for fMRI Prediction Tasks

Code: https://github.com/GonyRosenman/TFF.

Community-Aware Transformer



- **Method**
- **Problem Definition**

n_k: Number of nodes in k^{th} community

We first parcellate the brain into **N ROIs** based on a given atlas.

functional connectivity (FC) matrix is constructed by calculating **the Pearson correlation coefficients** between pairs of brain ROIs

Brain Graph

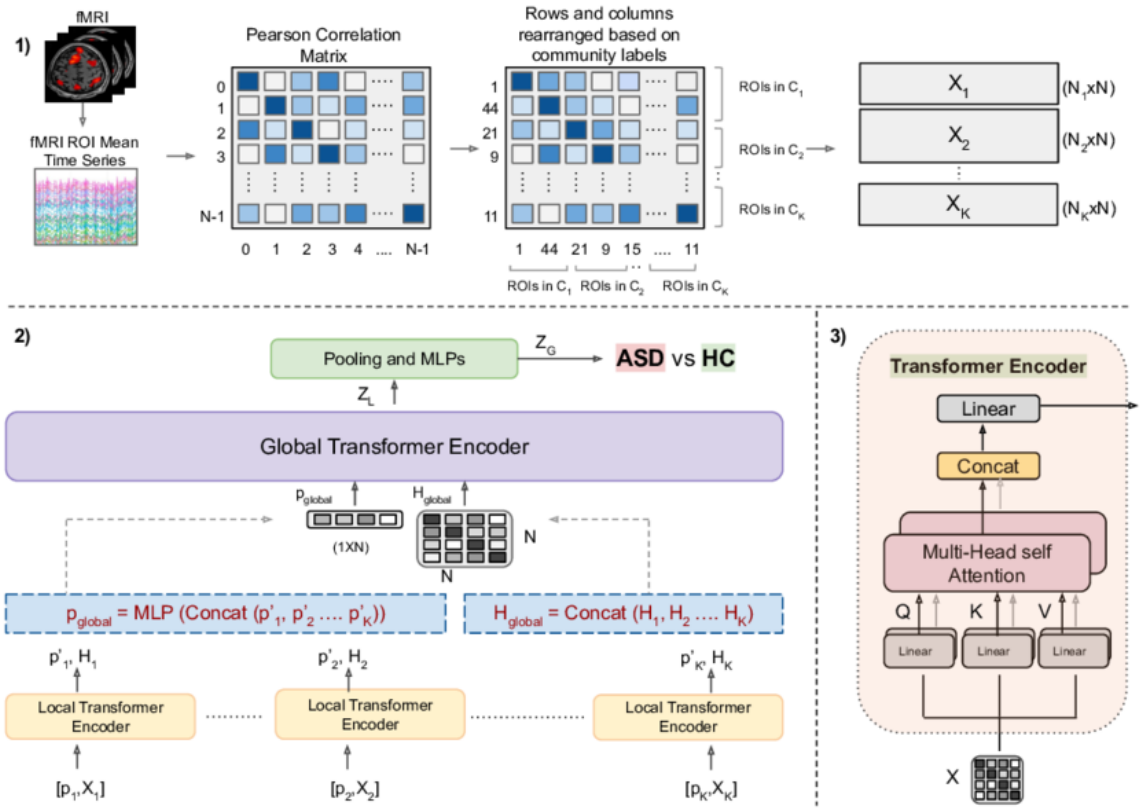
N: Number of nodes. $X \in R^{N \times N}$: FC matrix.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_j \\ \vdots \\ X_N \end{bmatrix}, X_j = [x_{j1} \quad \cdots \quad x_{jN}]: \text{Node Feature Vector}$$

K: Number of functional communities

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_k \\ \vdots \\ X_K \end{bmatrix}, X_k = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \cdots & \vdots \\ x_{k1} & \cdots & x_{kN} \end{bmatrix}$$

This process helps in grouping together regions with similar functional connectivity patterns, facilitating the analysis of inter-community and intra-community connections.



X_k → **K local transformer** → Token $H_k \in R^{n_k \times N}$

$$H = \begin{bmatrix} H_1 \\ \vdots \\ H_K \end{bmatrix}$$

→ **Global Transformer** → $Z_L \in R^{N \times N}$ →

Pooling → **MLP** → Z_G → **Predict Output**

Appendix

Local-global transformer encoder

The transformer encoder takes as input an FC matrix and has a multi-head-attention module to capture interdependencies between nodes using attention mechanism.

The learnt node feature matrix H_i

$$H_i = \left(\prod_{m=1}^{M} h^m \right) W_o$$

M = number of attention heads

Transformer encoder module

$$X_i \in R^{N \times N_i}$$

$$h^m = softmax\left(\frac{Q^m (K^m)^T}{\sqrt{d_k^m}} \right) V^m, Q^m = W_Q X_i', K^m = W_K X_i', V^m = W_V X_i', X_i' = [p_i, X_i],$$

Personalied Prompt Tokens: $\{p_1, \ldots, p_K\}, p_i \in R^{N \times 1}$, W_Q, W_K, W_V are model matrix parameters.

Local Transformer

- For optimal analysis of fMRI brain connectomes, it is important to incorporate both functional community information and node features.

- Therefore, we introduce the knowledge of community labels to the network by grouping node features based on community labels producing inputs $\{X_1 \quad \cdots \quad X_K\}$.

- Use the same local transformer for all inputs, but introduce unique learnable personalized prompt tokens $\{p_1 \quad \cdots \quad p_K\}$, where $p_i \in R^{1\times N}$ that learn to distinguish between node feature matrices of each community.

- Using attention mechanism, pairwise connection strength between ROIs is learnt, producing community-specific node embeddings H_i and prompt tokens p_i.

$[p_i' \quad H_i]$ =LocalTransformer $([p_i \quad X_i]), i \in [1, \ldots, K]$

Global Transformer

- On obtaining community-specific node embeddings and prompt tokens from the local transformer, it is essential **to combine this information and design a module to learn the brain network at a global level**.
- Therefore, we introduce a global transformer encoder to learn inter-community dependencies. **Input to the global transformer is the concatenated, learnt node feature matrices from the local transformer and a prompt token.**
- Prompt tokens learnt by the local transformer contain valuable information to distinguish different communities and thus are concatenated and passed through an MLP to obtain a prompt token input for the global transformer as follows:

$$p_{global} = MLP\left(Concat\left(p'_1, \ldots, p'_K\right)\right), \qquad H_{global} = Concat(H_1, \ldots, H_K)$$

$$[p', Z_L] = \text{GlobalTransformer}\left(\left[p_{global}, H_{global}\right]\right)$$

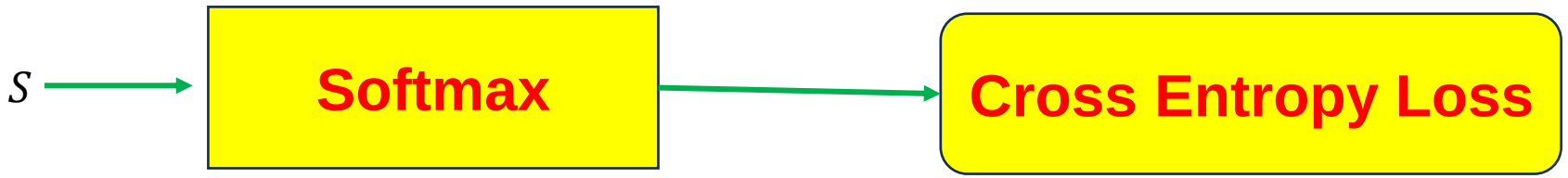
node embedding matrix: $Z_L \in R^{N \times N}$.

Cross Entropy Loss

- The most widely used loss function in machine learning applications is cross entropy.
- Given a true distribution t and a predicted distribution p, the cross entropy between them is given by the following equation:

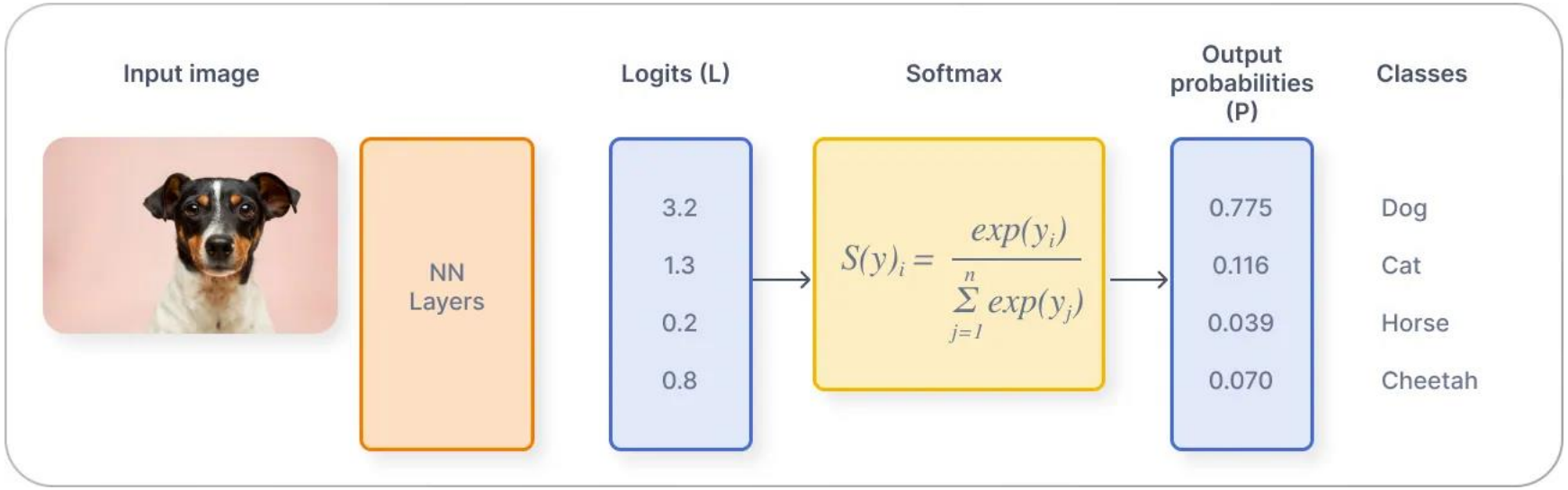
$$\boldsymbol{H(t,p)} = -\sum_{\boldsymbol{s\in S}} t(s)\log p(s)$$

- Cross-entropy (CE) loss functions



$$p(s_i) = \frac{\exp(s_i)}{\sum_{j=1}^{C}\exp(s_j)}$$

$$L_{CE}(t,p) = -\sum_{i=1}^{C} t_i \log p(s_i)$$



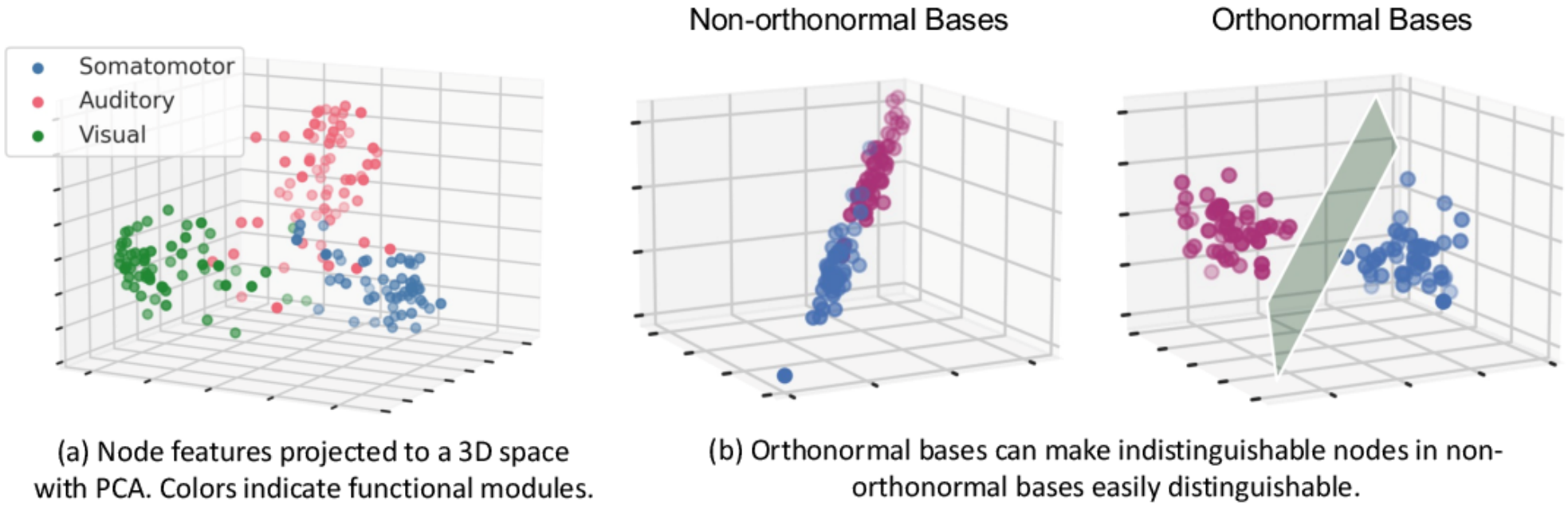
$$\overset{t}{\begin{bmatrix}1\\0\\0\\0\end{bmatrix}} \xrightarrow{\text{CE(t,p)}} \overset{p}{\begin{bmatrix}0.775\\0.116\\0.039\\0.070\end{bmatrix}}$$

$$L_{CE}(t,p) = -\log 0.775$$

Deval Shah, 2023. Cross Entropy Loss: Intro, Applications, Code

Graph Readout Layer

- The readout function is an essential component to learn the graph-level representations for brain network analysis (e.g., classification), which maps a set of learned node-level embeddings to a graph level embedding.
- **ORTHONORMAL CLUSTERING READOUT (OCREAD)**



(a) Node features projected to a 3D space with PCA. Colors indicate functional modules.

(b) Orthonormal bases can make indistinguishable nodes in non-orthonormal bases easily distinguishable.

Figure 1: Illustration of the motivations behind ORTHONORMAL CLUSTERING READOUT

Nodes in the same functional modules tend to have similar behaviors and clustered representations, as shown in Figure 1(a)

- Mean(·), Sum(·) and Max(·) are the most commonly used readout functions for GNNs.
- GNNs equipped with Sum(·) readout have the same discriminative power as the Weisfeiler-Lehman Test
- Zhang et al. propose a sort pooling to generate the graph-level representation by sorting the final node representations.
- Ju et al. present a layer-wise readout by extending the node information aggregated from the last layer of GNNs to all layers

- **However, none of the existing readout functions leverages the properties of brain networks that nodes in the same functional modules tend to have similar behaviors and clustered representations, as shown in Figure 1(a).**

- **ORTHONORMAL CLUSTERING READOUT**, where the graph-level embeddings are **pooled from clusters of functionally similar nodes through soft clustering with orthonormal projection.**

Step 1

Specifically, we first devise a self-supervised mechanism to jointly assign soft clusters to brain regions while learning their individual embeddings.

Step 2

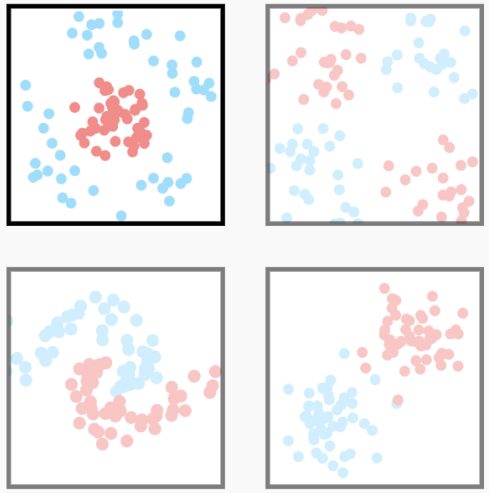
To further facilitate the learning of clusters and embeddings, we design an orthonormal projection

- **To address this deficiency, we design a novel readout function to take advantage of the modular-level similarities between ROIs in brain networks, where nodes are assigned softly to well-chosen clusters with an unsupervised process.**

Initialization

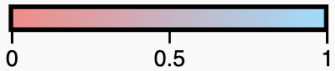
1. Choose input dataset

Select a training dataset.

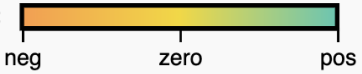


This legend details the color scheme for labels, and the values of the weights/gradients.

Label/Prediction:



Weight/Gradient:

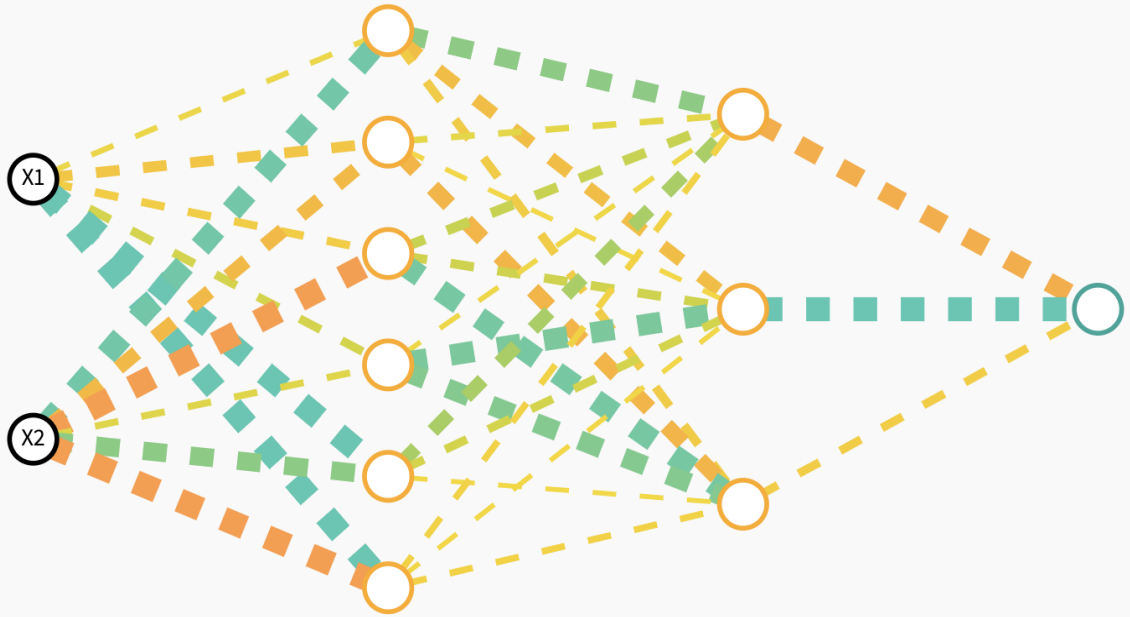


Node Type: Input Relu Sigmoid

2. Choose initialization method

Select an initialization method for the values of your neural network parameters[1].

Zero Too small Appropriate Too large

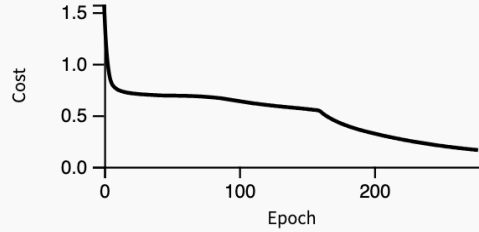


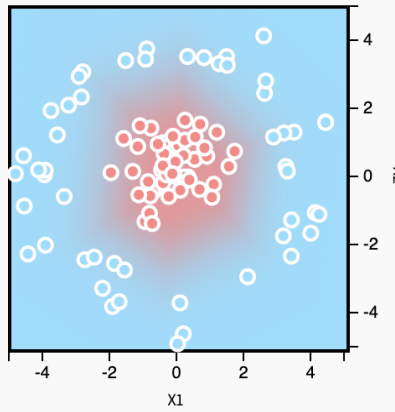
Select whether to visualize the weights or gradients of the network above.

Weight Gradient

3. Train the network.

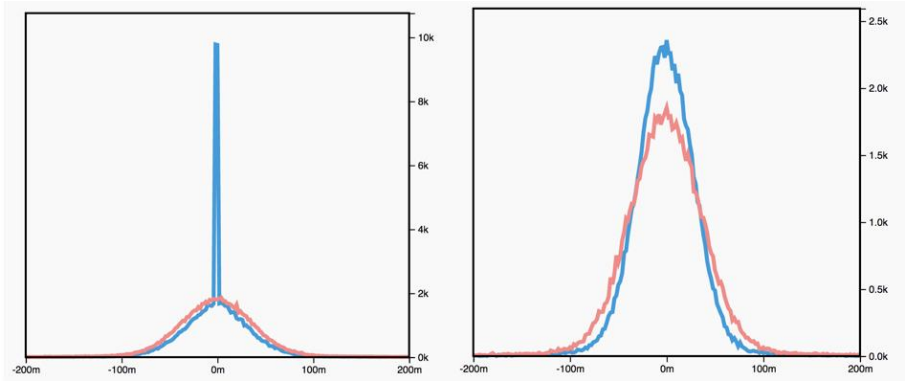
Observe the cost function and the decision bounda





Orthonormal initialization

- **Xavier Initialization, or Glorot Initialization, is an initialization scheme for neural networks. Biases are initialized be 0 and the weights at each layer are initialized as:**



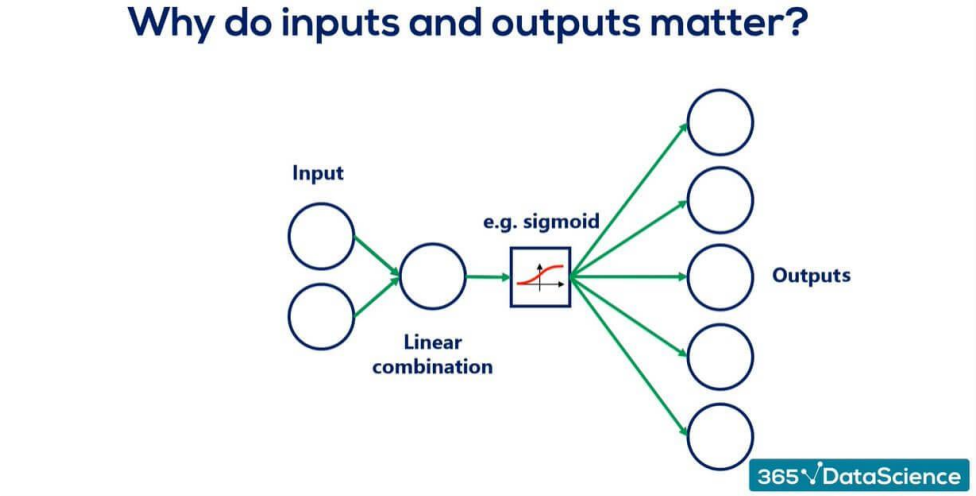
- **Uniform Xavier Initialization**

$$W_{ij} \sim U(-r, r), r = \sqrt{\frac{6}{n^{in}} + \frac{6}{n^{out}}}$$

- **Normal Xavier Initialization**

$$W_{ij} \sim N\left(0, \frac{2}{n^{in} + n^{out}}\right)$$

number of input n^{in} and output units n^{out} of each layer



Orthonormal Clustering Readout (OCRED) Procedures

1. Xavier uniform initialization to initialize K random centers, each center contains N dimensions, with weight matrix $C \in R^{K \times N}$

$$C = \begin{bmatrix} c_{11} & \cdots & C_{1N} \\ \vdots & \vdots & \vdots \\ C_{K1} & \cdots & C_{KN} \end{bmatrix}, c_{ij} \sim U(-r_i, r_i), r_i = \sqrt{\frac{6}{n_i^{out}}}$$

2. Apply the Gram-Schmidt process to obtain the orthonormal bases $E \in R^{K \times N}$

$$u_k = C_{k.} - \sum_{j=1}^{k-1} \frac{< u_j, C_{k.} >}{< u_j, u_j >} u_j, E_k = \frac{u_k}{\|u_k\|}$$

Appendix

3. A Softmax projection operator is used as the function to calculate the probability P_{ik} of assigning node i to cluster K:

$$P_{ik} = \frac{exp < Z_i^L, E_{k.} >}{\sum_{j=1}^{K} exp < Z_i^L, E_{j.} >}, P \in R^{N \times K}$$

$Z^L \in R^{N \times N}$ is the learned set of node embeddings from the last layer of MHSA module.

4. Calculate graph-level embedding Z_G:

$$Z_G = P^T Z^L. \qquad Z_G \in R^{K \times N}$$

Z_G is then flattened and passed to an MLP for a graph-level prediction with cross entropy loss function.

$$L_{CE}(t, p) = -\sum_{i=1}^{C} t_i \log p(s_i)$$

t_i: Indicator variable for the true class i.

$P(s_i)$: estimated probability for class s_i

Real Data Analysis

- **Datasets**

ABIDE (The Autism Brain Imaging Data Exchange) is a collection of resting-state functional MRI (rs-fMRI) data from **17 international sites** [with Configurable Pipeline for the Analysis of Connectomes (CPAC), band-pass filtering (0.01 - 0.1 Hz), no global signal regression, parcellated by Craddock 200 atlas.

ROIs of ABIDE dataset belong to either of **the eight functional communities** namely, cerebellum and subcortical structures (CS & SB), visual network (V), somatomotor network (SMN), DAN, ventral attention network (VAN), limbic network (L), frontoparietal network (FPN) and DMN.

ABIDE has fMRI data of **1009 subjects** **51.14% of** whom were diagnosed with **ASD**. Pearson correlation matrix is computed using mean time series of ROIs.

- **Experimental Settings:**

All models are implemented in **PyTorch** and trained on **NVIDIA V100** with 8GB memory. For both local and global transformers, the number of attention heads is equal to the number of communities. They used Adam optimizer with an initial learning rate of 10−4 and a weight decay of 10−4.

The train/validation/test data spilt ratio is **70:10:20**, and **the batch size is 64**. Prediction of ASD vs. HC is the binary classification task.

They use AUROC, accuracy, sensitivity, and specificity on the test set to evaluate performance. Models are trained for 50 epochs and they use an early stopping strategy. The epoch with the highest AUROC performance on the validation set is used for performance comparison on the test set.

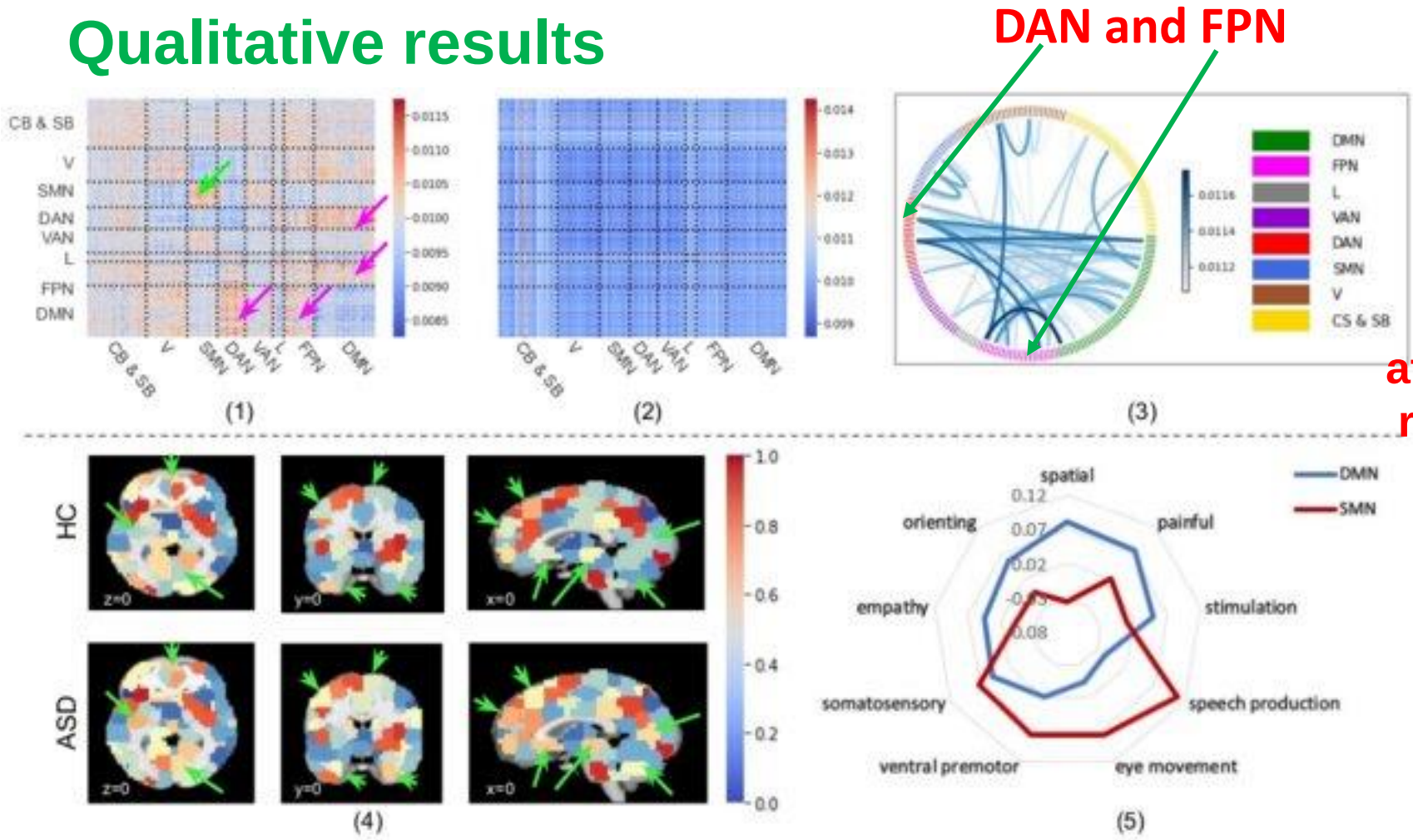
- ## Quantitative and Qualitative Results

Compare the performance of our model with three types of baselines (i) Comparison with transformer-based models BNT (ii) Comparison with neural network model on fixed brain network - BrainNetCNN[17] (iii) Comparison with neural network models on learnable brain network – FBNETGEN.

Table 1. Performance comparison with baselines

Model	AUROC	Accuracy	Sensitivity	Specifity
Brain Net	78.3	68.1	78.1	58.9
Brain NetCNN	74.1	67.5	65.3	69.6
FBNETGNN	72.5	64.9	60.9	67.5
Com-BrainTF	79.6	72.58	80.1	65.7

Qualitative results



These attention matrices are obtained by averaging attention scores over correctly classified test data.

attention scores in the SMN region (green arrow) are high

attention scores in DMN, DAN, and FPN regions were found to be high (magenta arrows)

Fig. 2. (1) **Attention matrix** of the global transformer of Com-BrainTF highlighting prominent communities that influence ASD prediction (2) Attention matrix of the first transformer encoder module of BNT (3) Chord plot of the top 1% of attention values (4) Neurosynth results generated from averaged prompt vectors (5) Correlation between functional communities and functional keywords decoded by Neurosynth

Ablation Study

Table 2. Ablation study: Examining different input and output possibilities

Different inputs to the global transformer				
Inputs	AUROC	Accuracy	Sensitivity	Specificity
Prompt tokens only	73.2	65.7	88.8	43.7
Node features and prompt token	79.6	72.5	80.1	65.7
CE loss on node features vs prompt token				
Outputs	AUROC	Accuracy	Sensitivity	Specificity
Prompt token	71.9	66.1	80.4	51.5
Node features	79.6	72.5	8.1	65.7

Table.2 reveals that prompt tokens alone are unable to capture the complete intra-functional connectivity and negatively affect performance.

The results (Table.2) demonstrated a significant decrease in performance when prompt token alone was used. This is expected since brain networks are non-euclidean structures and the learnt node features capture more information about the underlying graph, making them essential for accurately capturing the inter-community dependencies

PTGB: Pre-Train Graph Neural Networks for Brain Network Analysis

The full implementation of this work is publicly available at https://github.com/Owen-Yang-18/BrainNN-PreTrain.

- **Three real-world brain network datasets**:

1) the Bipolar Disorder (BP) dataset,
2) the Human Immunodeciency Virus Infection (HIV) dataset,
3) the Parkinson's Progression Markers Initiative (PPMI) dataset

while the large-scale PPMI dataset is publicly available for authorized users

https://www.ppmi-info.org/

- **Data preprocessing pipelines provided by the opensource BrainGB platform**

https://braingb.us/

MOLE-BERT: RETHINKING PRE-TRAINING GRAPH NEURAL NETWORKS FOR MOLECULES