BRAIN NETWORK TRANSFORMER

Xuan Kan[1] **Wei Dai**[2] **Hejie Cui**[1] **Zilong Zhang**[3] **Ying Guo**[1] **Carl Yang**[1]
[1]Emory University [2]Stanford University [3]University of International Business and Economics
{xuan.kan,hejie.cui,yguo2,j.carlyang}@emory.edu
dvd.ai@stanford.edu 201957020@uibe.edu.cn

Abstract

Human brains are commonly modeled as networks of Regions of Interest (ROIs) and their connections for the understanding of brain functions and mental disorders. Recently, Transformer-based models have been studied over different types of data, including graphs, shown to bring performance gains widely. In this work, we study Transformer-based models for brain network analysis. Driven by the unique properties of data, we model brain networks as graphs with nodes of fixed size and order, which allows us to (1) use connection profiles as node features to provide natural and low-cost positional information and (2) learn pairwise connection strengths among ROIs with efficient attention weights across individuals that are predictive towards downstream analysis tasks. Moreover, we propose an ORTHONORMAL CLUSTERING READOUT operation based on self-supervised soft clustering and orthonormal projection. This design accounts for the underlying functional modules that determine similar behaviors among groups of ROIs, leading to distinguishable cluster-aware node embeddings and informative graph embeddings. Finally, we re-standardize the evaluation pipeline on the only one publicly available large-scale brain network dataset of ABIDE, to enable meaningful comparison of different models. Experiment results show clear improvements of our proposed BRAIN NETWORK TRANSFORMER on both the public ABIDE and our restricted ABCD datasets. The implementation is available at https://github.com/Wayfear/BrainNetworkTransformer.

1 Introduction

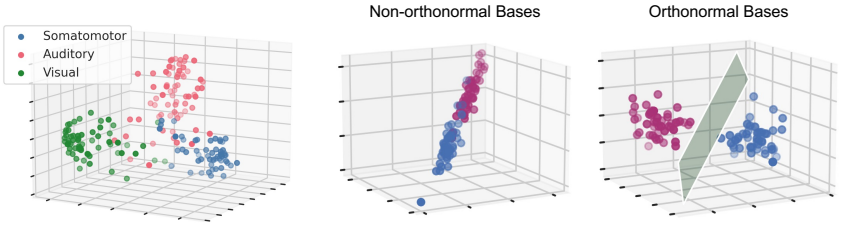
Brain network analysis has been an intriguing pursuit for neuroscientists to understand human brain organizations and predict clinical outcomes [50, 59, 58, 5, 18, 27, 52, 29, 58, 28, 41, 44, 31]. Among various neuroimaging modalities, functional Magnetic Resonance Imaging (fMRI) is one of the most commonly used for brain network construction, where the nodes are defined as Regions of Interest (ROIs) given an atlas, and the edges are calculated as pairwise correlations between the blood-oxygen-level-dependent (BOLD) signal series extracted from each region [54, 53, 59, 16]. Researchers observe that some regions can co-activate or co-deactivate simultaneously when performing cognitive-related tasks such as action, language, and vision. Based on this pattern, brain regions can be classified into diverse functional modules to analyze diseases towards their diagnosis, progress understanding and treatment.

Nowadays Transformer-based models have led a tremendous success in various downstream tasks across fields including natural language processing [56, 17] and computer vision [20, 10, 55]. Recent efforts have also emerged to apply Transformer-based designs to graph representation learning. GAT [57] firstly adapts the attention mechanism to graph neural networks (GNNs) but only considers the local structures of neighboring nodes. Graph Transformer [21] injects edge information into the attention mechanism and leverages the eigenvectors of each node as positional embeddings. SAN [40] further enhances the positional embeddings by considering both eigenvalues and eigenvectors and improves the attention mechanism by extending the attention from local to global structures.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Graphomer [64], which achieves the first place on the quantum prediction track of OGB Large-Scale Challenge [30], designs unique mechanisms for molecule graphs such as centrality encoding to enhance node features and spatial/edge encoding to adapt attention scores.

However, brain networks have several unique traits that make directly applying existing graph Transformer models impractical. First, one of the simplest and most frequently used methods to construct a brain network in the neuroimaging community is via pairwise correlations between BOLD time courses from two ROIs [43, 35, 13, 63, 69]. This impedes the designs like centrality, spatial, and edge encoding because each node in the brain network has the same degree and connects to every other node by a single hop. Second, in previous graph transformer models, eigenvalues and eigenvectors are commonly used as positional embeddings because they can provide identity and positional information for each node [15, 26]. Nevertheless, in brain networks, the connection profile, which is defined as each node's corresponding row in the brain network adjacency matrix, is recognized as the most effective node feature [13]. This node feature naturally encodes both structural and positional information, making the aforementioned positional embedding design based on eigenvalues and eigenvectors redundant. The third challenge is scalability. Typically, the numbers of nodes and edges in molecule graphs are less than 50 and 2500, respectively. However, for brain networks, the node number is generally around 100 to 400, while the edge number can be up to 160,000. Therefore, operations like the generation of all edge features in existing graph transformer models can be time-consuming, if not infeasible.



(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.

In this work, we propose to develop BRAIN NETWORK TRANSFORMER (BRAINNETTF), which leverages the unique properties of brain network data to fully unleash the power of Transformer-based models for brain network analysis. Specifically, motivated by previous findings on effective GNN designs for brain networks [13], we propose to use the effective initial node features of connection profiles. Empirical analysis shows that connection profiles naturally provide positional features for Transformer-based models and avoid the costly computations of eigenvalues or eigenvectors. Moreover, recent work demonstrates that GNNs trained on learnable graph structures can achieve superior effectiveness and explainability [35]. Inspired by this insight, we propose to learn fully pairwise attention weights with Transformer-based models, which resembles the process of learning predictive brain network structures towards downstream tasks.

One step further, when GNNs are used for brain network analysis, a graph-level embedding needs to be generated through a readout function based on the learned node embeddings [37, 43, 13]. As is shown in Figure 1(a), a property of brain networks is that brain regions (nodes) belonging to the same functional modules often share similar behaviors regarding activations and deactivations in response to various stimulations [7]. Unfortunately, the current labeling of functional modules is rather empirical and far from accurate. For example, [3] provides more than 100 different functional module organizations based on hierarchical clustering. In order to leverage the natural functions of brain regions without the limitation of inaccurate functional module labels, we design a new global pooling operator, ORTHONORMAL CLUSTERING READOUT, where the graph-level embeddings are pooled from clusters of functionally similar nodes through soft clustering with orthonormal projection. Specifically, we first devise a self-supervised mechanism based on [60] to jointly assign soft clusters to brain regions while learning their individual embeddings. To further facilitate the learning of clusters and embeddings, we design an orthonormal projection and theoretically prove its effectiveness in distinguishing embeddings across clusters, thus obtaining expressive graph-level embeddings after the global pooling, as illustrated in Figure 1(b).

Finally, the lack of open-access datasets has been a non-negligible challenge for brain network analysis. The strict access restrictions and complicated extraction/preprocessing of brain networks from fMRI data limit the development of machine learning models for brain network analysis. Specifically, among all the large-scale publicly available fMRI datasets in literature, ABIDE [6] is the only one provided with extracted brain networks fully accessible without permission requirements. However, ABIDE is aggregated from 17 international sites with different scanners and acquisition parameters. This inter-site variability conceals inter-group differences that are really meaningful, which is reflected in the unstable training performance and the significant gap between validation and testing performance in practice. To address these limitations, we propose to apply a stratified sampling method in the dataset splitting process and standardize a fair evaluation pipeline for meaningful model comparison on the ABIDE dataset. Our extensive experiments on this public ABIDE dataset and a restricted ABCD dataset [8] show significant improvements brought by our proposed BRAIN NETWORK TRANSFORMER.

2 Background and Related Work

2.1 GNNs for Brain Network Analysis

Recently, emerging attention has been devoted to the generalization of GNN-based models to brain network analysis [42, 2]. GroupINN [62] utilizes a grouping-based layer to provide explainability and reduce the model size. BrainGNN [43] designs the ROI-aware GNNs to leverage the functional information in brain networks and uses a special pooling operator to select these crucial nodes. IBGNN [14] proposes an interpretable framework to analyze disorder-specific ROIs and prominent connections. In addition, FBNetGen [35] considers the learnable generation of brain networks and explores the explainability of the generated brain networks towards downstream tasks. Another benchmark paper [13] systematically studies the effectiveness of various GNN designs over brain network data. Different from other work focusing on static brain networks, STAGIN [39] utilizes GNNs with spatio-temporal attention to model dynamic brain networks extracted from fMRI data.

2.2 Graph Transformer

Graph Transformer raises many researchers' interest currently due to its outstanding performance in graph representation learning. Graph Transformer [21] firstly injects edge information into the attention mechanism and leverages the eigenvectors as positional embeddings. SAN [40] enhances the positional embeddings and improves the attention mechanism by emphasizing neighbor nodes while incorporating the global information. Graphomer [64] designs unique mechanisms for molecule graphs and achieves the SOTA performance. Besides, a fine-grained attention mechanism is developed for node classification [68]. Also, the Transformer is extended to larger-scale heterogeneous graphs with a particular sampling algorithm in HGT [32]. EGT [33] further employs edge augmentation to assist global self-attention. In addition, LSPE [22] leverages the learnable structural and positional encoding to improve GNNs' representation power, and GRPE [49] enhances the design of encoding node relative position information in Transformer.

3 BRAIN NETWORK TRANSFORMER

3.1 Problem Definition

In brain network analysis, given a brain network $\boldsymbol{X} \in \mathbb{R}^{V \times V}$, where V is the number of nodes (ROIs), the model aims to make a prediction indicating biological sex, presence of a disease or other properties of the brain subject. The overall framework of our proposed BRAIN NETWORK TRANSFORMER is shown in Figure 2, which is mainly composed of two components, an L-layer attention module MHSA and a graph pooling operator OCREAD. Specifically, in the first component of MHSA, the model learns attention-enhanced node features $\boldsymbol{Z}^L$ through a non-linear mapping $\boldsymbol{X} \rightarrow \boldsymbol{Z}^L \in \mathbb{R}^{V \times V}$. Then the second component of OCREAD compresses the enhanced node embeddings $\boldsymbol{Z}^L$ to graph-level embeddings $\boldsymbol{Z}_G \in \mathbb{R}^{K \times V}$, where K is a hyperparameter representing the number of clusters. $\boldsymbol{Z}_G$ is then flattened and passed to a multi-layer perceptron for graph-level predictions. The whole training process is supervised with the cross-entropy loss.

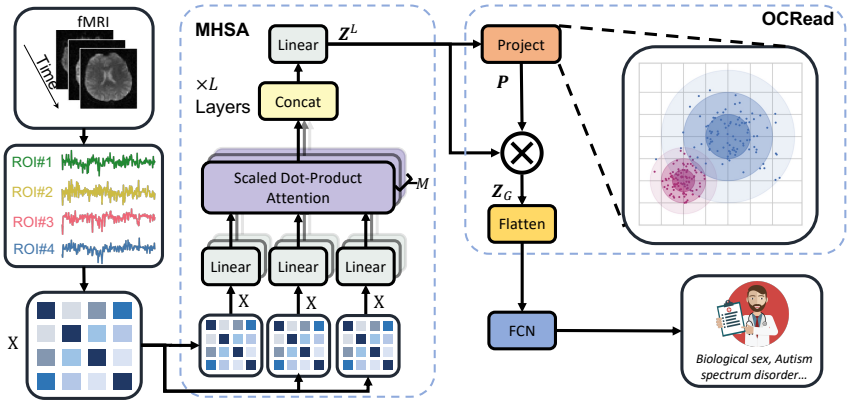


Figure 2: The overall framework of our proposed BRAIN NETWORK TRANSFORMER.

3.2 Multi-Head Self-Attention Module (MHSA)

To develop a powerful Transformer-based model suitable for brain networks, two fundamental designs, the positional embedding and attention mechanism, need to be reconsidered to fit the natural properties of brain network data. In existing graph transformer models, the positional information is usually encoded via eigendecomposition, while the attention mechanism often combines node positions with existing edges to calculate the attention scores. However, for the dense (often fully connected) graphs of brain networks, eigendecomposition is rather costly, and the existence of edges is hardly informative.

ROI node features on brain networks naturally contain sufficient positional information, making the positional embeddings based on eigendecomposition redundant. Previous work on brain network analysis has shown that the connection profile $\boldsymbol{X}_{i\cdot}$ for node i, defined as the corresponding row for each node in the edge weight matrix $\boldsymbol{X}$, always achieves superior performance over others such as node identities, degrees or eigenvector-based embeddings [43, 35, 13]. With this node feature initialization, the self-connection weight $\boldsymbol{x}_{ii}$ on the diagonal is always equal to one, which encodes sufficient information to determine the position of each node in a fully connected graph based on the given brain atlas. To verify this insight, we also empirically compare the performance of the original connection profile with two variants concatenated with additional positional information, *i.e.*, connection profile w/ identity feature and connection profile w/ eigen feature. The results indeed show no benefit brought by the additional computations (c.f. Appendix B). As for the attention mechanism, previous work [13] has empirically demonstrated that integrating edge weights into the attention score calculation can significantly degrade the effectiveness of attention on complete graphs, while the generation of edge-wise embedding can be unaffordable given a large number of edges in brain networks. On the other hand, the existence of edges provides no useful information for the computation of attention scores as well because all edges simply exist in complete graphs.

Based on the observations above, we design the basic BRAIN NETWORK TRANSFORMER by (1) adopting the connection profile as initial node features and eliminating any extra positional embeddings and (2) adopting the vanilla pair-wise attention mechanism without using edge weights or relative position information to learn a singular attention score for each edge in the complete graph.

Formally, we leverage a L-layer non-linear mapping module, namely Multi-Head Self-Attention (MHSA), to generate more expressive node features $\boldsymbol{Z}^L = \mathrm{MHSA}(\boldsymbol{X}) \in \mathbb{R}^{V\times V}$. For each layer l, the output $\boldsymbol{Z}^l$ is obtained by

$$\boldsymbol{Z}^l = (\|_{m=1}^{M} \boldsymbol{h}^{l,m})\boldsymbol{W}_{\mathcal{O}}^l, \boldsymbol{h}^{l,m} = \mathrm{Softmax}\left(\frac{\boldsymbol{W}_{\mathcal{Q}}^{l,m}\boldsymbol{Z}^{l-1}(\boldsymbol{W}_{\mathcal{K}}^{l,m}\boldsymbol{Z}^{l-1})^\top}{\sqrt{d_{\mathcal{K}}^{l,m}}}\right)\boldsymbol{W}_{\mathcal{V}}^{l,m}\boldsymbol{Z}^{l-1}, \quad (1)$$

where $\boldsymbol{Z}^0 = \boldsymbol{X}$, $\|$ is the concatenation operator, M is the number of heads, l is the layer index, $\boldsymbol{W}_{\mathcal{O}}^l, \boldsymbol{W}_{\mathcal{Q}}^{l,m}, \boldsymbol{W}_{\mathcal{K}}^{l,m}, \boldsymbol{W}_{\mathcal{V}}^{l,m}$ are learnable model parameters, and $d_{\mathcal{K}}^{l,m}$ is the first dimension of $\boldsymbol{W}_{\mathcal{K}}^{l,m}$.

3.3 ORTHONORMAL CLUSTERING READOUT (OCREAD)

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. $\mathrm{Mean}(\cdot), \mathrm{Sum}(\cdot)$ and $\mathrm{Max}(\cdot)$ are the most commonly used readout functions for GNNs. Xu et al. [61] show that GNNs equipped with $\mathrm{Sum}(\cdot)$ readout have the same discriminative power as the Weisfeiler-Lehman Test. Zhang et al. [66] propose a sort pooling to generate the graph-level representation by sorting the final node representations. Ju et al. [34] 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). 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.

Formally, given K cluster centers, each center has V dimensions, $\boldsymbol{E} \in \mathbb{R}^{K \times V}$, a Softmax projection operator is used as the function to calculate the probability $\boldsymbol{P}_{ik}$ of assigning node i to cluster k,

$$\boldsymbol{P}_{ik} = \frac{e^{\langle \boldsymbol{Z}^L_{i\cdot}, \boldsymbol{E}_{k\cdot} \rangle}}{\sum_{k'}^{K} e^{\langle \boldsymbol{Z}^L_{i\cdot}, \boldsymbol{E}_{k'\cdot} \rangle}}, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product and $\boldsymbol{Z}^L$ is the learned set of node embeddings from the last layer of MHSA module. With this computed soft assignment $\boldsymbol{P} \in \mathbb{R}^{V \times K}$, the original learned node representation $\boldsymbol{Z}^L$ can be aggregated under the guidance of the soft cluster information, where the graph-level embedding $\boldsymbol{Z}_G$ is obtained by $\boldsymbol{Z}_G = \boldsymbol{P}^\top \boldsymbol{Z}^L$.

However, jointly learning node embeddings and clusters without ground-truth cluster labels is difficult. To obtain representative soft assignment $\boldsymbol{P}$, the initialization of K cluster centers $\boldsymbol{E}$ is critical and should be designed delicately. To this end, we leverage the observation illustrated in Figure 1(b), where orthonormal embeddings can improve the clustering of nodes in brain networks *w.r.t.* the functional modules underlying brain regions.

Orthonormal Initialization. To initialize a group of orthonormal bases as cluster centers, we first adopt the Xavier uniform initialization [25] to initialize K random centers and each center contains V dimensions $\boldsymbol{C} \in \mathbb{R}^{K \times V}$. Then, we apply the Gram-Schmidt process to obtain the orthonormal bases $\boldsymbol{E}$, where

$$\boldsymbol{u}_k = \boldsymbol{C}_{k\cdot} - \sum_{j=1}^{k-1} \frac{\langle \boldsymbol{u}_j, \boldsymbol{C}_{k\cdot} \rangle}{\langle \boldsymbol{u}_j, \boldsymbol{u}_j \rangle} \boldsymbol{u}_j, \quad \boldsymbol{E}_{k\cdot} = \frac{\boldsymbol{u}_k}{\|\boldsymbol{u}_k\|}. \tag{3}$$

In the next section, we theoretically prove the advantage of this orthonormal initialization.

3.3.1 Theoretical Justifications

In OCREAD, proper cluster centers can generate higher-quality soft assignments and enlarge the difference between $\boldsymbol{P}$ from different classes. [51, 46] showed the advantages of orthogonal initialization in DNN model parameters. However, none of them proves whether it is an ideal strategy to obtain the cluster centers. We propose two methods from the perspective of statistics as follows.

Firstly, to discern features of different nodes, we would expect a larger discrepancy among their similarity probabilities indicated from the readout. One way to measure the discrepancy is using the *variance* of $\boldsymbol{P}$ for each feature. Let $\bar{\boldsymbol{P}} \equiv 1/K$ denote the mean of any discrete probabilities with K values. Variance of $\boldsymbol{P}$ measures the difference between $\boldsymbol{P}$ and $\bar{\boldsymbol{P}}$. We average over the feature vector space: if the result is small, then there is a large tendency that different $\boldsymbol{P}$ approaches $\bar{\boldsymbol{P}}$ and hence cannot be discerned easily. Specifically, the following theorem holds for our function Eq. (2):

Theorem 3.1. *For arbitrary $r > 0$, let $B_r = \{\mathcal{Z} \in \mathbb{R}^V; \|\mathcal{Z}\| \leq r\}$ denote the round ball centered at origin of radius r with $\mathcal{Z}$ being fracture vectors. Let V_r be the volume of B_r. The variance of Softmax projection averaged over B_r*

$$\frac{1}{V_r} \int_{B_r} \sum_{k}^{K} \Big(\frac{e^{\langle \mathcal{Z}, \boldsymbol{E}_{k\cdot} \rangle}}{\sum_{k'}^{K} e^{\langle \mathcal{Z}, \boldsymbol{E}_{k'\cdot} \rangle}} - \frac{1}{K} \Big)^2 d\mathcal{Z}, \tag{4}$$

attains maximum when $\boldsymbol{E}$ is orthonormal.

Despite the concise form, it is unclear whether the above integral has an elementary antiderivative. Even though, we can circumvent this problem and a rigorous proof is given in Appendix C.

The second statistical method shows that for general readout functions without a known analytical form, initializing with orthonormal cluster centers has a larger probability of gaining better performance. To set up the proper statistical scenario, we assume that the unknown readout is obtained by a regression of some samples $(\hat{\boldsymbol{Z}}^{(s)}, \hat{\boldsymbol{E}}^{(t)}, \hat{\boldsymbol{P}}^{(st)})$. This formally converts the exact functional relationship between $\boldsymbol{Z}_{i\cdot}, \boldsymbol{E}_{k\cdot}$ and $\boldsymbol{P}_{ik}$ to a *statistical relationship*:

$$\boldsymbol{P}_T(\boldsymbol{Z}_{i\cdot}, \boldsymbol{E}_{k\cdot}) = \boldsymbol{P}(\boldsymbol{Z}_{i\cdot}, \boldsymbol{E}_{k\cdot}) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), \quad E(\epsilon_i) = 0, \quad D(\epsilon_i) = \sigma^2, \tag{5}$$

with $\boldsymbol{P}_T$ being the probability *truly* reflecting similarities between nodes and clusters and ϵ_i denoting the stochastic error. It is almost impossible to find $\boldsymbol{P}_T$, but by computing the so-called *variation inflation factor* [47], we show that regression in orthonormal case has a higher accuracy than that in non-orthonormal case. Combining with a hypothesis testing, we obtain the following

Theorem 3.2. *The significance level $\alpha_{E_{k\cdot}}$ which reveals the probability of rejecting a well-estimated pooling is lower when sampling from orthonormal centers than that from non-orthonormal centers.*

More details can be seen in Appendix C.

3.4 Generalizing OCREAD to Other Graph Tasks and Domains

In this work, we tested the proposed OCREAD on functional connectivity (FC) based brain networks. Other popular modalities of brain networks include structural connectivities (SC), which describe the anatomical organization of the brain by measuring the fiber tracts between brain regions [4]. In SC-based brain networks, ROIs that are positionally close to each other on the structural connectivity networks tend to share similar connection profiles. This means the idea of OCREAD is also naturally applicable to SC networks, where the orthonormal clustering is based on the physical distances instead of the functional modules on FC.

At a higher level, the idea of our proposed OCREAD is not confined to graph-level prediction tasks on brain networks but can also be generalized to other graph learning tasks and domains. Precisely, there is a growing tendency in node/edge level prediction tasks to enhance the node/edge representation learning by utilizing the subgraph embeddings around each target node/edge [67, 65]. In this process, substructure learning needs to be performed on the subgraphs, where our proposed OCREAD can be adapted for compressing a set of node embeddings to subgraph embeddings. Besides, OCREAD is also potentially useful for other types of graphs in the biomedical domains. For example, for protein-protein interaction networks, proteins can be implicitly grouped by families that share common evolutionary origins [48], whereas for gene expression networks, genes can be grouped based on the latent pathway information [36]. Both of them are potential directions for the future application of OCREAD, among many others driven by biological or other types of prior knowledge regarding underlying node/edge groups.

4 Experiments

This section evaluates the effectiveness of our proposed BRAIN NETWORK TRANSFORMER (BRAINNETTF) with extensive experiments. We aim to address the following research questions:

RQ1. How does BRAINNETTF perform compared with state-of-the-art models of various types?

RQ2. How does our proposed OCREAD module perform with different model choices?

RQ3. Does the learned model of BRAINNETTF exhibit consistency with existing neuroscience knowledge and suggest reasonable explainability?

4.1 Experimental Settings

Datasets. We conduct experiments on two real-world fMRI datasets. (a) *Autism Brain Imaging Data Exchange (ABIDE)*: This dataset collects resting-state functional magnetic resonance imaging (rs-fMRI) data from 17 international sites, and all data are anonymous [6]. The used dataset contains brain networks from 1009 subjects, with 516 (51.14%) being Autism spectrum disorder

(ASD) patients (positives). The region definition is based on Craddock 200 atlas [12]. As the most convenient open-source large-scale dataset, it provides generated brain networks and can be downloaded directly without permission request. Despite the ease of acquisition, the heterogeneity of the data collection process hinders its use. Since multi-site data are collected from different scanners with different acquisition parameters, non-neural inter-site variability may mask inter-group differences. In practice, we find the training unstable, and there is a significant gap between validation and testing performances. However, we discover that most models can achieve a stable performance if we follow an appropriate stratified sampling strategy by considering collection sites during the training-validation-testing splitting process for ABIDE. Training curves in Appendix A also show how different models achieve a stabler performance on our designed new splitting settings than the random splitting. Therefore, we use ABIDE as one of the benchmark datasets in this work, and we share our re-standardized data splitting to provide a fair evaluation pipeline for various future methods. (b) *Adolescent Brain Cognitive Development Study (ABCD)*: This is one of the largest publicly available fMRI datasets with restricted access (a strict data requesting process needs to be followed to obtain the data) [8]. The data we use in the experiments are fully anonymized brain networks with only biological sex labels. After the quality control process, 7901 subjects are included in the analysis, with 3961 (50.1%) among them being female. The region definition is based on the HCP 360 ROI atlas [24].

Metrics. The diagnosis of ASD is the prediction target on ABIDE, while biological sex prediction is used as the evaluation task for ABCD. Both prediction tasks are binary classification problems, and both datasets are balanced between classes. Hence, AUROC is a proper performance metric adopted for fair comparison at various threshold settings, and accuracy is applied to reflect the prediction performance when the threshold is 0.5. Besides, since the model is mainly for medical applications, we add two critical metrics for diagnostic tests, Sensitivity and Specificity, which respectively refer to true positive rate and true negative rate. All reported performances are the average of 5 random runs on the test set with the standard deviation.

Implementation details. For experiments, we use a two-layer Multi-Head Self-Attention Module and set the number of heads M to 4 for each layer. We randomly split 70% of the datasets for training, 10% for validation, and the remaining are utilized as the test set. In the training process of BRAINNETTF, we use an Adam optimizer with an initial learning rate of 10^{-4} and a weight decay of 10^{-4}. The batch size is set as 64. All models are trained for 200 epochs, and the epoch with the highest AUROC performance on the validation set is used for performance comparison on the test set. The model is trained on an NVIDIA Quadro RTX 8000. Please refer to the repository and Appendix G for the full implementation of BRAINNETTF.

Computation complexity. In BRAINNETTF, the computation complexity of Multi-Head Self-Attention Module and OCREAD are $\mathcal{O}(LMV^2)$ and $\mathcal{O}(KV)$ respectively, where L is the layer number of Multi-Head Self-Attention Module, V is the number of nodes, M is the number of heads, and K is the number of clusters in OCREAD. The overall computation complexity of BRAINNETTF is thus $\mathcal{O}(V^2)$, which is on the same scale as common GNNs on brain networks such as BrainGNN [43] and BrainGB [13].

4.2 Performance Analysis (RQ1)

We compare FBNETGEN with baselines of three types. The details about how to tune hyperparameters of various baselines can be found in Appendix F. Besides, Appendix E shows the comparison of the number of parameters between our model and other baseline models, which shows that the parameter size of BRAINNETTF is larger than GNN and CNN models but smaller than other transformer models. **(a) BRAINNETTF vs. other graph transformers.** We compare BRAINNETTF with two popular graph Transformers, SAN [40] and Graphormer [64]. In addition, we also include a basic version of BRAINNETTF without OCREAD, composed of a Transformer with a 2-layer Multi-Head Self-Attention and a CONCAT-based readout named VanillaTF. Our BRAINNETTF outperforms SAN and Graphormer by significant margins, with up to 6% absolute improvements on both datasets. VanillaTF also surpasses SAN and Graphormer. We believe this downgraded performance of existing graph transformers results from their design flaws facing the natures of brain networks. Specifically, both the preprocessing and the training stages of the Graphormer model accepts only discrete, categorical data. A bin operator has to be applied on the adjacency matrix, coarsening the node feature from connection profiles and dramatically hurting the performance.

Furthermore, since brain networks are complete graphs, key designs like centrality encoding and spatial encoding of Graphormer cannot be appropriately applied. Similarly, for SAN, experiments in Appendix B show that adding eigen node features to connection profiles cannot improve the model's performance. Besides, the benchmark paper [13] reveals that injecting edge weights into the attention mechanism can significantly reduce the prediction power. Furthermore, Appendix D shows our BRAINNETTF is much faster than other graph transformers due to special optimizations towards brain networks. **(b) BRAINNETTF vs. neural network models on fixed brain networks.** We further introduce another three neural network baselines on fixed brain networks. BrainGNN [43] designs ROI-aware GNNs for brain network analysis. BrainGB [13] is a systematic study of how to design effective GNNs for brain network analysis. We adopt their best design as the BrainGB baseline. BrainnetCNN [37] represents state-of-the-art of specialized GNNs for brain network analysis, which models the adjacency matrix of a brain network similarly as a 2D image. As is shown in Table 1, BRAINNETTF consistently outperforms BrainGNN, BrainGB and BrainnetCNN. **(c) BRAINNETTF vs. neural network models on learnable brain networks.** Unlike classical GNNs, FBNETGEN [35], DGM [38] and BrainNetGNN [45] hold a similar idea, which is to apply GNNs based on a learnable graph. FBNETGEN achieves SOTA performance on the ABCD dataset for biological sex prediction, and the learnable graphs can be seen as a type of attention score. Experiment results show that our proposed BRAINNETTF beats all three of them on both datasets.

Table 1: Performance comparison with different baselines (%). The performance gains of BRAINNETTF over the baselines have passed the t-test with p-value<0.03.

Type	Method	Dataset: ABIDE				Dataset: ABCD			
		AUROC	Accuracy	Sensitivity	Specificity	AUROC	Accuracy	Sensitivity	Specificity
Graph Transformer	SAN	71.3±2.1	65.3±2.9	55.4±9.2	68.3±7.5	90.1±1.2	81.0±1.3	84.9±3.5	77.5±4.1
	Graphormer	63.5±3.7	60.8±2.7	**78.7±22.3**	36.7±23.5	89.0±1.4	80.2±1.3	81.8±11.6	82.4±7.4
	VanillaTF	76.4±1.2	65.2±1.2	66.4±11.4	71.1±12.0	94.3±0.7	85.9±1.4	87.7±2.4	82.6±3.9
Fixed Network	BrainGNN	62.4±3.5	59.4±2.3	36.7±24.0	70.7±19.3	OOM	OOM	OOM	OOM
	BrainGB	69.7±3.3	63.6±1.9	63.7±8.3	60.4±10.1	91.9±0.3	83.1±0.5	84.6±4.3	81.5±3.9
	BrainNetCNN	74.9±2.4	67.8±2.7	63.8±9.7	71.0±10.2	93.5±0.3	85.7±0.8	87.9±3.4	83.0±4.4
Learnable Network	FBNETGNN	75.6±1.2	68.0±1.4	64.7±8.7	62.4±9.2	94.5±0.7	87.2±1.2	87.0±2.5	86.7±2.8
	BrainNetGNN	55.3±1.9	51.2±5.4	67.7±37.5	33.9±34.2	75.3±5.2	67.5±4.7	67.7±5.7	68.0±6.5
	DGM	52.7±3.8	60.7±12.6	53.8±41.2	51.1±40.9	76.8±19.0	68.6±8.1	40.5±29.7	**95.6±4.2**
Ours	BRAINNETTF	**80.2±1.0**	**71.0±1.2**	72.5±5.2	**69.3±6.5**	**96.2±0.3**	**88.4±0.4**	**89.4±2.6**	88.4±1.5

4.3 Ablation Studies on the OCREAD Module (RQ2)

4.3.1 OCREAD with varying readout functions

We vary the readout function for various Transformer architectures, including SAN, Graphormer and VanillaTF, to observe the performance of each ablated model variant. The results shown in Table 2 demonstrate that our OCREAD is the most effective readout function for brain networks and improves the prediction power across various Transformer architectures.

Table 2: Performance comparison AUROC (%) with different readout functions.

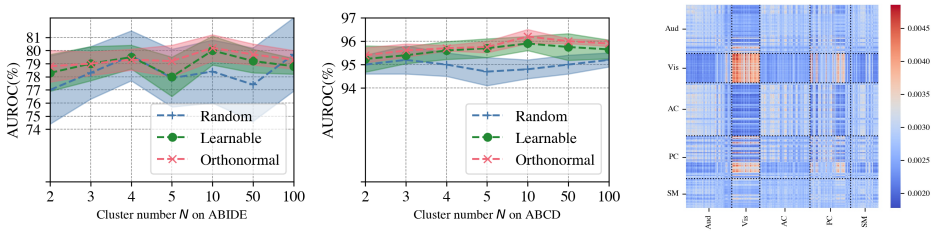
Readout	Dataset: ABIDE			Dataset: ABCD		
	SAN	Graphormer	VanillaTF	SAN	Graphormer	VanillaTF
MEAN	63.7±2.4	50.1±1.1	73.4±1.4	88.5±0.9	87.6±1.3	91.3±0.7
MAX	61.9±2.5	54.5±3.6	75.6±1.4	87.4±1.1	81.6±0.8	94.4±0.6
SUM	62.0±2.3	54.1±1.3	70.3±1.6	84.2±0.8	71.5±0.9	91.6±0.6
SortPooling	68.7±2.3	51.3±2.2	72.4±1.3	84.6±1.1	86.7±1.0	89.9±0.6
DiffPool	57.4±5.2	50.5±4.7	62.9±7.3	78.1±1.5	70.0±1.9	83.9±1.3
CONCAT	**71.3±2.1**	63.5±3.7	76.4±1.2	90.1±1.2	89.0±1.4	94.3±0.7
OCREAD	70.6±2.4	**64.9±2.7**	**80.2±1.0**	**91.2±0.7**	**90.2±0.7**	**96.2±0.4**

4.3.2 OCREAD with varying cluster initializations

To further demonstrate how the design of OCREAD influences the performance of BRAINNETTF, we investigate two key model selections, the initialization method for cluster centers and the cluster

number K. For the initialization, three different kinds of initialization procedures are compared, namely (a) **Random**: the Xavier uniform [25] is leveraged to randomly generate a group of centers, which are then normalized into unit vectors; (b) **Learnable**: the same initial process as Random, but the generated centers are further updated with gradient descent; (c) **Orthonormal**: our proposed process as described in Eq. (3).

Specifically, we test each initialization method with the cluster number K equals to 2, 3, 4, 5, 10, 50, 100. The results of adjusting these two hyper-parameters on ABIDE and ABCD datasets are shown in Figure 3(a). We observe that: (1) When cluster centers are orthonormal, the model's performance increases with the number of clusters ranging from 2 to 10, and then drops with the cluster number rising from 10 to 100, suggesting the optimal cluster number to be relatively small, which leads to less computation and is consistent with the fact that the typical number of functional modules are smaller than 25; (2) With a sufficiently large cluster number, all three initialization methods, Random, Learnable and Orthonormal, tend to reach similar performance, but orthonormal performs stably better when the number of clusters is smaller; (3) It is also notable that our OCREAD consistently achieves the best performance over other initialization methods regarding smaller standard deviations.



(a) Influence of two key hyper-parameters for model performance.

(b) Attention heatmap on ABCD.

Figure 3: The hyper-parameter influence and the heatmap from self-attention.

4.4 In-depth Analysis of Attention Scores and Cluster Assignments (RQ3)

Figure 3(b) displays the self-attention score from the first layer of Multi-Head Self-Attention. The attention scores are the average across all subjects in the ABCD test set. This figure shows that the learned attention scores well match the divisions of functional modules based on available labels, demonstrating the effectiveness and explainability of our Transformer model. Note that since there exists no available functional module labels for the atlas of the ABIDE dataset, we cannot visualize the correlations between attention scores and functional modules.

Figure 4 shows the cluster soft assignment results $\boldsymbol{P}$ on nodes in OCREAD with two initialization methods. The cluster number K is set to 4. The visualized numerical values are the average $\boldsymbol{P}$ of all subjects in each dataset's test set. From the visualization, we observe that (a) Base on Appendix H, orthonormal initialization produces more discriminative $\boldsymbol{P}$ between classes than random initialization; (b) Within each class, orthonormal initialization encourages the nodes to form groups. These observations demonstrate that our OCREAD with orthonormal initialization can leverage potential clusters underlying node embeddings, thus automatically grouping brain regions into potential functional modules.

5 Discussion and Conclusion

Neuroimaging technologies, including functional magnetic resonance imaging (fMRI) are powerful noninvasive tools for examining the brain functioning. There is an emerging nation-wide interest in conducting neuroimaging studies for investigating the connection between the biology of the brain, and demographic variables and clinical outcomes such as mental disorders. Such studies provide an unprecedented opportunity for cross-cutting investigations that may offer new insights to the differences in brain function and organization across subpopulations in the society (such as biological sex and age groups) as well as reveal neurophysiological mechanisms underlying brain disorders (such as psychiatric illnesses and neurodegenerative diseases). These studies have a tremendous impact in social studies and biomedical sciences. For example, mental disorders are the leading cause of disability in the USA and roughly 1 in 17 have a seriously debilitating mental illness. To

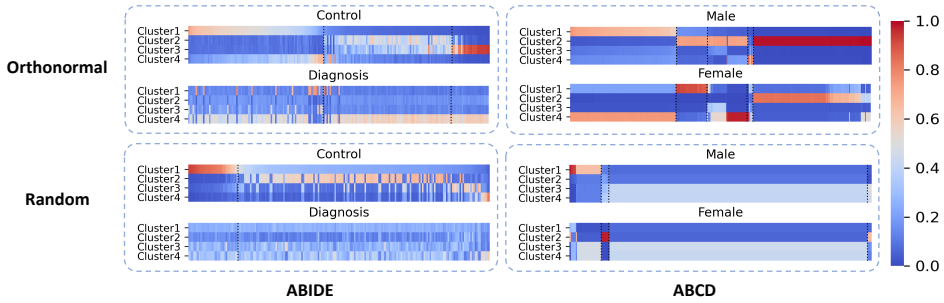


Figure 4: Visualization of cluster (module-level) embeddings learned with Orthonormal vs. Random cluster center initializations on two datasets. Each group in the dotted box contains two heatmaps (one for each prediction class) with the same node ordering on the x-axis.

address this burden, national institutions such as NIH have included brain-behavior research as one of their strategic objectives and stated that sound efforts must be made to redefine mental disorders into dimensions or components of observable behaviors that are more closely aligned with the biology of the brain. Using brain imaging data to predict diagnosis has great potential to result in mechanisms that target for more effective preemption and treatment.

In this paper, we presents BRAIN NETWORK TRANSFORMER, a specialized graph Transformer model with ORTHONORMAL CLUSTERING READOUT for brain network analysis. Extensive experiments on two large-scale brain network datasets demonstrate that our BRAINNETTF achieves superior performance over SOTA baselines of various types. Specifically, to model the potential node feature similarity in brain networks, we design OCREAD and prove its effectiveness both theoretically and empirically. Lastly, the re-standardized dataset split for ABIDE can provide a fair evaluation for new methods in the community. For future work, BRAINNETTF can be improved with explicit explanation modules and used as the backbone for further brain network analysis, such as digging essential neural circuits for mental disorders and understanding cognitive development in adolescents.

6 Acknowledgments

This research was supported in part by the University Research Committee of Emory University, and the internal funding and GPU servers provided by the Computer Science Department of Emory University. The authors gratefully acknowledge support from NIH under award number R01MH105561 and R01MH118771. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Data used in the preparation of this article were obtained from the Adolescent Brain Cognitive Development (ABCD) Study (`https://abcdstudy.org`), held in the NIMH Data Archive (NDA). This is a multisite, longitudinal study designed to recruit more than 10,000 children age 9-10 and follow them over 10 years into early adulthood. The ABCD Study® is supported by the National Institutes of Health and additional federal partners under award numbers U01DA041048, U01DA050989, U01DA051016, U01DA041022, U01DA051018, U01DA051037, U01DA050987, U01DA041174, U01DA041106, U01DA041117, U01DA041028, U01DA041134, U01DA050988, U01DA051039, U01DA041156, U01DA041025, U01DA041120, U01DA051038, U01DA041148, U01DA041093, U01DA041089, U24DA041123, U24DA041147. A full list of supporters is available at `https://abcdstudy.org/federal-partners.html`. A listing of participating sites and a complete listing of the study investigators can be found at `https://abcdstudy.org/consortium_members/`. ABCD consortium investigators designed and implemented the study and/or provided data but did not necessarily participate in the analysis or writing of this report. This manuscript reflects the views of the authors and may not reflect the opinions or views of the NIH or ABCD consortium investigators. The ABCD data repository grows and changes over time. The ABCD data used in this report came from NIMH Data Archive Release 4.0 (DOI 10.15154/1523041). DOIs can be found at `https://nda.nih.gov/abcd`.

References

[1] G. A. F. and C. J. Wild. *Nonlinear Regression: Seber/Nonlinear Regression*. 1989.

[2] David Ahmedt-Aristizabal, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. Graph-based deep learning for medical diagnosis and analysis: past, present and future. *Sensors*, 21:4758, 2021.

[3] Teddy J Akiki and Chadi G Abdallah. Determining the hierarchical architecture of the human brain using subject-level clustering of functional networks. *Scientific Reports*, 9:1–15, 2019.

[4] Parinaz Babaeeghazvini, Laura M. Rueda-Delgado, Jolien Gooijers, Stephan P. Swinnen, and Andreas Daffertshofer. Brain structural and functional connectivity: A review of combined works of diffusion magnetic resonance imaging and electro-encephalography. *Frontiers in Human Neuroscience*, 15, 2021.

[5] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10:186–198, 2009.

[6] Craddock Cameron, Benhajali Yassine, Chu Carlton, Chouinard Francois, E. Aykan Alan, Jakab András, Khundrakpam Budhachandra, Lewis John, Liub Qingyang, Milham Michael, Yan Chaogan, and Bellec Pierre. The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives. *Frontiers in Neuroinformatics*, 2013.

[7] Alfonso Caramazza and Max Coltheart. Cognitive neuropsychology twenty years on. *Cognitive Neuropsychology*, 2006.

[8] B.J. Casey, Tariq Cannonier, and May I. Conley et al. The adolescent brain cognitive development (abcd) study: Imaging acquisition across 21 sites. *Developmental Cognitive Neuroscience*, 32:43–54, 2018.

[9] Gemai Chen and N. Balakrishnan. A General Purpose Approximate Goodness-of-Fit Test. *Journal of Quality Technology*, 27:154–161, 1995.

[10] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS*, 2021.

[11] A. Colin Cameron and Frank A.G. Windmeijer. An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics*, 77:329–342, 1997.

[12] R Cameron Craddock, G Andrew James, Paul E Holtzheimer III, Xiaoping P Hu, and Helen S Mayberg. A whole brain fmri atlas generated via spatially constrained spectral clustering. *Human Brain Mapping*, 33:1914–1928, 2012.

[13] Hejie Cui, Wei Dai, Yanqiao Zhu, Xuan Kan, Antonio Aodong Chen Gu, Joshua Lukemire, Liang Zhan, Lifang He, Ying Guo, and Carl Yang. Braingb: A benchmark for brain network analysis with graph neural networks, 2022.

[14] Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. Interpretable graph neural networks for connectome-based brain disorder analysis. In *MICCAI*, 2022.

[15] Hejie Cui, Zijie Lu, Pan Li, and Carl Yang. On positional and structural node features for graph neural networks on non-attributed graphs. *CIKM*, 2022.

[16] Tian Dai, Ying Guo, Alzheimer's Disease Neuroimaging Initiative, et al. Predicting individual brain functional connectivity using a bayesian hierarchical model. *NeuroImage*, 147:772–787, 2017.

[17] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*, 2019.

[18] Gustavo Deco, Viktor K. Jirsa, and Anthony R. McIntosh. Emerging concepts for the dynamical organization of resting-state activity in the brain. *Nature Reviews Neuroscience*, 12:43–56, 2011.

[19] Morris H. DeGroot and Mark J. Schervish. *Probability and Statistics*. 4th ed edition, 2012.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[21] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

[22] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *ICLR*, 2022.

[23] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[24] Matthew F. Glasser, Stamatios N. Sotiropoulos, J. Anthony Wilson, Timothy S. Coalson, Bruce Fischl, Jesper L. Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R. Polimeni, David C. Van Essen, and Mark Jenkinson. The minimal preprocessing pipelines for the human connectome project. *NeuroImage*, 80:105–124, 2013.

[25] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[26] Shupeng Gui, Xiangliang Zhang, Pan Zhong, Shuang Qiu, Mingrui Wu, Jieping Ye, Zhengdao Wang, and Ji Liu. Pine: Universal deep embedding for graph nodes via partial permutation invariant set functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:770–782, 2022.

[27] Ying Guo and Giuseppe Pagnoni. A unified framework for group independent component analysis for multi-subject fmri data. *NeuroImage*, 42(3):1078–1093, 2008.

[28] Ixavier A Higgins, Suprateek Kundu, Ki Sueng Choi, Helen S Mayberg, and Ying Guo. A difference degree test for comparing brain networks. *Human brain mapping*, pages 4518–4536, 2019.

[29] Ixavier A Higgins, Suprateek Kundu, and Ying Guo. Integrative bayesian analysis of brain functional networks incorporating anatomical knowledge. *Neuroimage*, 181:263–278, 2018.

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

[31] Yingtian Hu, Mahmoud Zeydabadinezhad, Longchuan Li, and Ying Guo. A multimodal multilevel neuroimaging model for investigating brain connectome development. *Journal of the American Statistical Association*, pages 1–15, 2022.

[32] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *WWW*, pages 2704–2710, 2020.

[33] Md Shamim Hussain, Mohammed J Zaki, and Dharmashankar Subramanian. Edge-augmented graph transformers: Global self-attention is enough for graphs. *arXiv*, 2021.

[34] Mingxuan Ju, Shifu Hou, Yujie Fan, Jianan Zhao, Liang Zhao, and Yanfang Ye. Adaptive kernel graph neural network. *AAAI*, 2022.

[35] Xuan Kan, Hejie Cui, Joshua Lukemire, Ying Guo, and Carl Yang. FBNETGEN: Task-aware GNN-based fMRI analysis via functional brain network generation. In *Medical Imaging with Deep Learning*, 2022.

[36] Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28:27–30, 2000.

[37] Jeremy Kawahara, Colin J. Brown, Steven P. Miller, Brian G. Booth, Vann Chau, Ruth E. Grunau, Jill G. Zwicker, and Ghassan Hamarneh. BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *NeuroImage*, 146:1038–1049, 2017.

[38] Anees Kazi, Luca Cosmo, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Differentiable graph module (dgm) for graph convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022.

[39] Byung-Hoon Kim, Jong Chul Ye, and Jae-Jin Kim. Learning dynamic graph representation of brain connectome with spatio-temporal attention. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *NeurIPS*, 2021.

[40] Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *NeurIPS*, 2021.

[41] Suprateek Kundu, Joshua Lukemire, Yikai Wang, and Ying Guo. A novel joint brain network analysis using longitudinal alzheimer's disease data. *Scientific reports*, 9(1):1–18, 2019.

[42] Xiaoxiao Li, Nicha C. Dvornek, Yuan Zhou, Juntang Zhuang, Pamela Ventola, and James S. Duncan. Graph neural network for interpreting task-fmri biomarkers. In *MICCAI*, 2019.

[43] Xiaoxiao Li, Yuan Zhou, Siyuan Gao, Nicha Dvornek, Muhan Zhang, Juntang Zhuang, Shi Gu, Dustin Scheinost, Lawrence Staib, Pamela Ventola, et al. Braingnn: Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 2021.

[44] Joshua Lukemire, Suprateek Kundu, Giuseppe Pagnoni, and Ying Guo. Bayesian joint modeling of multiple brain functional networks. *Journal of the American Statistical Association*, 116:518–530, 2021.

[45] Usman Mahmood, Zening Fu, Vince D. Calhoun, and Sergey Plis. A deep learning model for data-driven discovery of functional connectivity. *Algorithms*, 14, 2021.

[46] Haitao Mao, Xu Chen, Qiang Fu, Lun Du, Shi Han, and Dongmei Zhang. *Neuron Campaign for Initialization Guided by Information Bottleneck Theory*. 2021.

[47] F. H. C. Marriott, J. Neter, W. Wasserman, and M. H. Kutner. Applied Linear Regression Models. *Biometrics*, 1985.

[48] Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik L L Sonnhammer, Silvio C E Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, Robert D Finn, and Alex Bateman. Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49:D412–D419, 2020.

[49] Wonpyo Park, Woonggi Chang, Donggeon Lee, Juntae Kim, and Seung-won Hwang. Grpe: Relative positional encoding for graph transformer, 2022.

[50] Theodore D. Satterthwaite, Daniel H. Wolf, David R. Roalf, Kosha Ruparel, Guray Erus, Simon Vandekar, Efstathios D. Gennatas, Mark A. Elliott, Alex Smith, Hakon Hakonarson, Ragini Verma, Christos Davatzikos, Raquel E. Gur, and Ruben C. Gur. Linked Sex Differences in Cognition and Functional Connectivity in Youth. *Cerebral Cortex*, 25:2383–2394, 2015.

[51] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. 2014.

[52] Ran Shi and Ying Guo. Investigating differences in brain functional networks using hierarchical covariate-adjusted independent component analysis. *The annals of applied statistics*, page 1930, 2016.

[53] Sean L Simpson, F DuBois Bowman, and Paul J Laurienti. Analyzing complex functional brain networks: fusing statistics and network science to understand the brain. *Statistics Surveys*, 7:1, 2013.

[54] Stephen M. Smith, Karla L. Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F. Beckmann, Thomas E. Nichols, Joseph D. Ramsey, and Mark W. Woolrich. Network modelling methods for FMRI. *NeuroImage*, 54, 2011.

[55] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *ECCV*, 2022.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NeurIPS*, 2017.

[57] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[58] Yikai Wang and Ying Guo. A hierarchical independent component analysis model for longitudinal neuroimaging studies. *NeuroImage*, 189:380–400, 2019.

[59] Yikai Wang, Jian Kang, Phebe B. Kemmer, and Ying Guo. An Efficient and Reliable Statistical Method for Estimating Functional Connectivity in Large Scale Brain Networks Using Partial Correlation. *Frontiers in Neuroscience*, 10:123, 2016.

[60] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016.

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

[62] Yujun Yan, Jiong Zhu, Marlena Duda, Eric Solarz, Chandra Sripada, and Danai Koutra. Groupinn: Grouping-based interpretable neural network-based classification of limited, noisy brain data. In *KDD*, 2019.

[63] Yi Yang, Yanqiao Zhu, Hejie Cui, Xuan Kan, Lifang He, Ying Guo, and Carl Yang. Data-efficient brain connectome analysis via multi-task meta-learning. *KDD*, 2022.

[64] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *NeurIPS*, 2021.

[65] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *NeurIPS*, 2018.

[66] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *AAAI*, 2018.

[67] Muhan Zhang and Pan Li. Nested graph neural networks. In *NeurIPS*, 2021.

[68] Jianan Zhao, Chaozhuo Li, Qianlong Wen, Yiqi Wang, Yuming Liu, Hao Sun, Xing Xie, and Yanfang Ye. Gophormer: Ego-graph transformer for node classification, 2021.

[69] Yanqiao Zhu, Hejie Cui, Lifang He, Lichao Sun, and Carl Yang. Joint embedding of structural and functional brain networks with graph neural networks for mental illness diagnosis. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 272–276, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Appendix C.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix C.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Implementation details in Section 4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Metrics in Section 4.1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Implementation details in Section 4.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Datasets in Section 4.1.
   (b) Did you mention the license of the assets? [N/A] ABIDE and ABCD do not provide any license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provide our implementation and share our repository with MIT license.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Datasets in Section 4.1.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Datasets in Section 4.1.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Training Curves of Different Models with or without StratifiedSampling

In Figure 5, we demonstrate the training curves of different models with or without stratified sampling based on site information from ABIDE. The curves of different variants display similar patterns across three model architectures in a single run. We remove Graphormer since its performance is much worse than others. Specifically, it is shown that (a) with stratified sampling, the performance gap between validation and test on ABIDE is much smaller than the one without stratified sampling; (b) stratified sampling can stabilize the training process on ABIDE, especially for VanillaTF and BRAINNETTF.

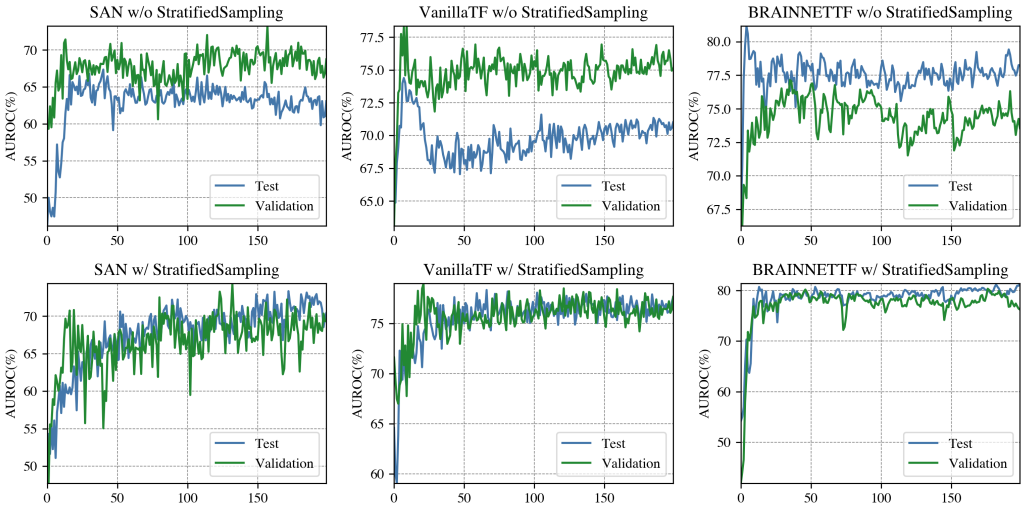


Figure 5: Training Curves of Different Models with or without StratifiedSampling.

B Transformer Performance with Different Node Features

We compare the performance of Transformer model equipped with different node features. The results are shown in Table 3, where connection profile represents the corresponding row for each node in the adjacency matrix, identity feature initializes a unique one-hot vector for each node, and eigen feature generates a k-dimensional feature vector for each node from the k eigenvectors based on the eigendecomposition on the adjacency matrix. Empirical observations demonstrate that adding identity or eigen node features to connection profiles cannot improve the model's performance.

Model	**Node Feature**	**Dataset**	
		ABIDE	ABCD
VanillaTF	Connection Profile	76.4±1.2	94.3±0.7
	Connection Profile w/ Identity Feature	75.4±1.9	94.5±0.6
	Connection Profile w/ Eigen Feature	75.9±2.1	94.0±0.8

Table 3: The Performance (AUROC%) of Transformer with Different Node Features.

C Statistical Proof of the Goodness with Orthonormal Cluster Centers

We propose two statistical methods to prove the goodness in orthonormal case since it is impractical to directly compare the performance of the orthonormal and non-orthonormal initializations.

C.1 Proof of Theorem 3.1

We state Theorem 3.1 here and show the proof details.

Theorem C.1. *For arbitrary $r > 0$, let $B_r = \{\mathcal{Z} \in \mathbb{R}^V; \|\mathcal{Z}\| \leq r\}$ denote the round ball centered at origin of radius r with $\mathcal{Z}$ being fracture vectors. Let V_r be the volume of B_r. The variance of Softmax projection averaged over B_r*

$$\frac{1}{V_r}\int_{B_r}\sum_k^K\Big(\frac{e^{\langle \mathcal{Z},\boldsymbol{E}_{k\cdot}\rangle}}{\sum_{k'}^K e^{\langle \mathcal{Z},\boldsymbol{E}_{k'\cdot}\rangle}}-\frac{1}{K}\Big)^2 d\mathcal{Z}, \tag{6}$$

attains maximum when $\boldsymbol{E}$ is orthonormal.

Proof. For simplicity, we first consider the two-dimensional case with two cluster centers $\boldsymbol{E}_1, \boldsymbol{E}_2$. Since we integrate over the round ball B_r, spherical symmetry allows us to set $\boldsymbol{E}_1 = (1, 0)$ and $\boldsymbol{E}_1 = (\cos(\phi), \sin(\phi))$ with $\phi \in [0, \frac{\pi}{21}]$ being the angle between $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ under polar coordinates. Then the Softmax readout Eq. (2) can be rewritten as:

$$\boldsymbol{P}_1 = \frac{e^{\rho\cos(\theta)}}{e^{\rho\cos(\theta)}+e^{\rho\cos(\theta-\phi)}}, \quad \boldsymbol{P}_2 = \frac{e^{\rho\cos(\theta-\phi)}}{e^{\rho\cos(\theta)}+e^{\rho\cos(\theta-\phi)}}, \tag{7}$$

where θ is the angle between $\mathcal{Z}$ and $\boldsymbol{E}_1$ and ρ is the norm of $\mathcal{Z}$. Hence, the integral is

$$F(\phi) := \frac{1}{V_r}\int_{B_r}\sum_{k=1}^2(\boldsymbol{P}_k-\frac{1}{2})^2 d\mathcal{Z} = \frac{1}{\pi r^2}\int_0^r\int_0^{2\pi}\Big(\frac{e^{2\rho\cos(\theta)}+e^{2\rho\cos(\theta-\phi)}}{(e^{\rho\cos(\theta)}+e^{\rho\cos(\theta-\phi)})^2}+\frac{1}{2}\Big)d\theta d\rho. \tag{8}$$

Our aim is to show that the integral $F(\phi)$ attains its maximum when $\boldsymbol{E}_1, \boldsymbol{E}_2$ are orthogonal. It is unclear whether the above integral has an elementary antiderivative. Thus, instead of evaluating the integral directly, we firstly prove two symmetric properties of the integrand $f(\rho, \theta, \phi)$: (a) It is straightforward to show that $f(\rho, \theta + k\pi, \phi) = f(\rho, \theta, \phi)$ for $k \in \mathbb{N}$. That is, f is periodic for π on the first argument θ. (b) We have

$$\begin{aligned} f(\frac{\phi}{2}+\frac{\pi}{2}-\theta) &= \frac{e^{2\rho\sin(\frac{\phi}{2}+\theta)}+e^{-2\rho\sin(\frac{\phi}{2}-\theta)}}{(e^{\rho\sin(\frac{\phi}{2}+\theta)}+e^{-\rho\sin(\frac{\phi}{2}-\theta)})^2} \\ &= \frac{e^{2\rho\sin(\frac{\phi}{2}+\theta)}+e^{-2\rho\sin(\frac{\phi}{2}-\theta)}}{e^{2\rho\sin(\frac{\phi}{2}+\theta)}+e^{-2\rho\sin(\frac{\phi}{2}-\theta)}+2e^{\rho\sin(\frac{\phi}{2}+\theta)-\rho\sin(\frac{\phi}{2}-\theta)}} \\ &= \frac{e^{2\rho\sin(\frac{\phi}{2}-\theta)}+e^{-2\rho\sin(\frac{\phi}{2}+\theta)}}{(e^{\rho\sin(\frac{\phi}{2}-\theta)}+e^{-\rho\sin(\frac{\phi}{2}+\theta)})^2} = f(\frac{\phi}{2}+\frac{\pi}{2}+\theta), \end{aligned} \tag{9}$$

which means f is symmetric with respect to $\theta = \frac{\phi}{2} + \frac{\pi}{2} + k\pi$. As the integrand $f(\rho, \theta, \phi)$ is periodic, we are allowed to compare $F(\phi_1), F(\phi_2)$ via

$$\begin{aligned} \int_{\frac{\phi_1}{2}}^{\frac{\phi_1}{2}+2\pi} f(\rho,\theta,\phi_1)d\theta &= \int_0^{2\pi} f(\rho,\theta,\phi_1)d\theta, \\ \int_{\frac{\phi_1}{2}}^{\frac{\phi_2}{2}+2\pi} f(\rho,\theta,\phi_2)d\theta &= \int_0^{2\pi} f(\rho,\theta,\phi_2)d\theta. \end{aligned} \tag{10}$$

The integral domain $[\frac{\phi}{2}, \frac{\phi}{2} + 2\pi]$ is taken according to the second symmetry property of f and can be significant for the following trick: we take the directional derivative of f along $\boldsymbol{v} = (1, 2)$ tangent to the straight line $\theta = \frac{\phi}{2}$:

$$\begin{aligned} Df(\boldsymbol{v}) &= \frac{\partial f}{\partial\theta}+2\frac{\partial f}{\partial\phi} \\ &= \frac{2\rho e^{\rho\cos(\theta-\phi)+\rho\cos(\theta)}(e^{\rho\cos(\theta-\phi)}-e^{\rho\cos(\theta)})(\sin(\theta)+\sin(\theta-\phi))}{(e^{\rho\cos(\theta-\phi)}+e^{\rho\cos(\phi)})^3}. \end{aligned} \tag{11}$$

It is easy to check that in the above integral domain and for any $\rho > 0$, $Df(\boldsymbol{v})$ is always non-negative. Hence,

$$f(\rho, \theta - \frac{\phi_1}{2}, \phi_1) \leq f(\rho, \theta - \frac{\phi_2}{2}, \phi_2) \tag{12}$$

when $\phi_1 \leq \phi_2$. After taking integral, $F(\phi_1) \leq F(\phi_2)$ and thus it attains maximum in the orthonormal case ($\phi = \frac{\pi}{2}$). Comparing $F(\phi_1), F(\phi_2)$ without adjusting the integral domain as above cannot give a clear result because the simple partial derivative $\partial f / \partial \phi$ oscillates around zero. Higher dimensional cases follow similarly by employing spherical and hyperspherical coordinates. □

C.2 Proof of Theorem 3.2

Theorem 3.2 deals with a more general case: comparing the performance of an arbitrary readout $\boldsymbol{P}$ defined by orthonormal cluster centers with non-orthonormal ones. We regard $\boldsymbol{P}$ as an estimated similarity probability between nodes and clusters and solve this problem from the perspective of statistics. The estimation is considered as a regression of samples $(\hat{\boldsymbol{Z}}^{(s)}, \hat{\boldsymbol{E}}^{(t)}, \hat{\boldsymbol{P}}^{(st)})$ from node features, cluster centers and similarity probabilities. We then judge the estimation relative to true similarity probability $\boldsymbol{P}_T$. Although it is almost impossible to find an analytic formula for $\boldsymbol{P}_T$, we can indirectly judge the quality of estimation. To clarify the idea, we introduce some basic concepts from statistics and prove our results on a statistical basis.

C.2.1 Background Knowledge of Regression Analysis

We first consider process samples by logistic regression with cluster centers as *categorical variables*. Intuitively, non-orthonormal centers correlate with each other, which means there is an *overlap* among categorical variables and makes it hard to identify the *decision boundary* that leads to a failed classification. However, as far as we know, it is *unclear* how to compare overlaps between orthonormal and non-orthonormal variables rigorously. Thus, we simply process samples by a general nonlinear regression. The regression process is linearized by the Gauss-Newton algorithm to facilitate the analysis. We judge the *goodness-of-fit* describing the degree to which the regression function fits its observed value, and then conduct a hypothesis test. The *goodness-of-fit* is measured by *coefficient of determinate* R^2 [47]:

Definition C.2. We consider a regression with r independent main variables:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_r X_r + \epsilon. \tag{13}$$

Let $\hat{x}_p = (\hat{x}_{p1}, ..., \hat{x}_{ps})^\top$ and $\hat{y} = (\hat{y}_1, ..., \hat{y}_s)^\top$ be data sets (samples) associated with *fitted values* $\check{y} = (\check{y}_1, ..., \check{y}_s)$. Each difference $e_q = \hat{y}_q - \check{y}_q$ is called a *residue*. We denote the mean of $\hat{x}_p$ and $\hat{y}$ by $\bar{x}_p, \bar{y}$. The variability of data set can be measured by the *total sum of squares* (SST), the *sum of squares of residuals* (SSR) and the *explained sum of squares* (SSE) defined as (where $p = 1, 2, ..., r$ $q = 1, 2, ..., s$):

$$\text{SST} = \sum_q (\hat{y}_q - \bar{y})^2, \quad \text{SSR} = \sum_q e_q^2 = \sum_q (\hat{y}_q - \check{y}_q)^2, \quad \text{SSE} = \sum_{q,p} (\hat{x}_{qp} - \bar{x}_p)^2. \tag{14}$$

In linear regression, SSR + SSE = SST and the coefficient of determination R^2 is defined as:

$$R^2 = \frac{\text{SSE}}{\text{SST}} = 1 - \frac{\text{SSR}}{\text{SST}}. \tag{15}$$

Conceptually, SSE is the error cost by regression of main variables. Thus by definition, R^2 reveals the percentage of errors that main variables can explain in the total error SST. The value of R^2 is bounded by 1. A large value of R^2 indicates a better fitting. However, it should be noted that an extremely-large R^2 could indicate overfitting.

In our problem, since our regression is nonlinear, the sum of SSR and SSE is less than SST [1]. Therefore, measuring *goodness-of-fit* by R^2 in nonlinear regression is inaccurate. A common strategy to remedy this problem is approximating nonlinear functions by polynomials via *Gauss-Newton algorithm*. We provide a brief introduction here, and more details can be found in [1]: for a nonlinear

model f_k with parameter δ, in a small neighborhood of δ_T-the true value of δ, we have the linear expansion:

$$f_k(\delta) \approx f_k(\delta_T) + \sum_{m=1}^{M} \frac{\partial f_k}{\partial \delta_m}\Big|_{\delta_T} (\delta_m - \delta_{Tm}). \tag{16}$$

Or briefly, we write it by *vector notation*:

$$\boldsymbol{f}(\delta) \approx \boldsymbol{f}(\delta_T) + \boldsymbol{F}(\delta - \delta_T), \tag{17}$$

where $\boldsymbol{F}(\delta - \delta_T)$ stands for the dot product of derivatives and differences of parameters from Eq. (16). Suppose $\delta^{(\gamma)}$ is an approximation to the least-squares estimation δ of our model, for δ close to $\delta^{(\gamma)}$, we rewrite the expansion as:

$$\check{\boldsymbol{P}} = \boldsymbol{f}(\delta) \approx \boldsymbol{f}(\delta^{(\gamma)}) + \boldsymbol{F}^{(\gamma)}(\delta - \delta^{(\gamma)}), \tag{18}$$

where $\check{\boldsymbol{P}}$ denotes a fitted value of $\boldsymbol{P}$ and $F^{(\gamma)}(\delta - \delta^{(\gamma)})$ again means a dot product. Applying this to the residual vector $\boldsymbol{e}(\delta)$, we have:

$$\boldsymbol{e}(\delta) = \boldsymbol{P} - \boldsymbol{f}(\delta) \approx \boldsymbol{e}(\delta^{(\gamma)}) - \boldsymbol{F}^{(\gamma)}(\delta - \delta^{(\gamma)}). \tag{19}$$

Thus, the norm

$$\begin{aligned} S(\delta) &:= \|\boldsymbol{P} - f(\delta)\|^2 = \boldsymbol{e}^\top(\delta)\boldsymbol{e}(\delta) \\ &\approx \boldsymbol{e}^\top(\delta^{(\gamma)})\boldsymbol{e}(\delta^{(\gamma)}) - 2\boldsymbol{e}^\top(\delta^{(\gamma)})\boldsymbol{F}^{(\gamma)}(\delta - \delta^{(\gamma)}) + (\delta - \delta^{(\gamma)})^\top \boldsymbol{F}^{(\gamma)\top}\boldsymbol{F}^{(\gamma)}(\delta - \delta^{(\gamma)}). \end{aligned} \tag{20}$$

The right-hand side is minimized with respect to δ when

$$\delta - \delta^{(\gamma)} = (\boldsymbol{F}^{(\gamma)\top}\boldsymbol{F}^{(\gamma)})^{-1}\boldsymbol{F}^{(\gamma)\top}\boldsymbol{e}(\delta^{(\gamma)}) = \zeta^{(\gamma)}. \tag{21}$$

This suggests that given a current approximation $\delta^{(\gamma)}$, the next approximation should be:

$$\delta^{(\gamma+1)} = \delta^{(\gamma)} + \zeta^{(\gamma)}. \tag{22}$$

Expanding the nonlinear function $\boldsymbol{f}$ as polynomials and modifying the parameter δ as above, we can use R^2 to measure the *goodness-of-fit*. To acquire higher accuracy in a general nonlinear regression, one can make a elaborated *goodness-of-fit test* for specific fitting functions e.g., [9, 11]. We do not discuss this sophisticated method as it is out of the scope of this paper.

C.2.2 Comparing R^2 by Variance Inflation Factor

The proof of Theorem 3.2 consists of two steps: (a) we first prove that the *regression accuracy*, the accuracy when regressing $\boldsymbol{P}$ is higher when sampling from orthonormal cluster centers (Theorem C.4), and consequently (b) higher regression accuracy increases *appraisal accuracy*, the accuracy when appraising an estimated value in *hypothesis testing* (Theorem C.6).

In this subsection, we compare regression accuracy. we fix $\boldsymbol{Z}_i$ when regressing $\boldsymbol{P}$ via the fitted value $\check{\boldsymbol{P}}(\boldsymbol{E}_k)$. Statistically, the expectation $\mathrm{E}(\boldsymbol{P})$ of all readouts is identified as the true similarly probability $\boldsymbol{P}_T$. In regression analysis, the Ordinary Least Squares (OLS) guarantees asymptotically unbiased estimations. That is, when the sample size s is large enough, it can be regarded as an *unbiased estimation* [47]:

$$\mathrm{E}(\check{\boldsymbol{P}}) = \boldsymbol{P}_T = \mathrm{E}(\boldsymbol{P}). \tag{23}$$

Therefore, the better the *goodness-of-fit* reflected by R^2, the smaller the variance of estimation. To compare this, we use the concept of *variance inflation factor* which reflects the inflation of weights of variables in regression:

Definition C.3. The variance inflation factor $(\mathrm{VIF})_p$ is defined as:

$$(\mathrm{VIF})_p = \frac{1}{(1 - R_p^2)}, \tag{24}$$

where R_p^2 is the coefficient of multiple determination when X_p is regressed by the r-1 other variables in the model from Eq. (13).

Remark. We discuss more details about VIF in the following context [47]. For simplicity, we denote the following collection of samples and regression coefficients:

$$\hat{X} = (\hat{x}_1, ..., \hat{x}_r) = (\hat{x}_{qp}), \quad \hat{y} = (\hat{y}_1, ..., \hat{y}_s)^\top, \quad \beta = (\beta_1, ..., \beta_r).$$

In the regression model Eq. (13), the estimation $\check{\beta}_p$ of regression coefficients β_p are obtained by Ordinary Least Squares (OLS):

$$\check{\beta} = (\hat{X}^\top \hat{X})^{-1} \hat{X}^\top \hat{y}. \tag{25}$$

We standardize the regression equation by covariance matrices σ_y of $\check{y}$ and the variance σ_q of $\hat{x}_p$ as

$$\check{y}_q^* = \frac{\check{y}_q - \bar{y}}{\sigma_y}, \quad \hat{x}_{qp}^* = \sigma_q^{-1}(\hat{x}_{pq} - \bar{x}_p), \tag{26}$$

and

$$\check{\beta}_q^* = \check{\beta}_q \frac{\sigma_q}{\sigma_y}, \quad \check{y}^* = \check{\beta}_0^* + \check{\beta}_1^* X_1^* + \check{\beta}_2^* X_2^* + \cdots + \check{\beta}_r^* X_r^*. \tag{27}$$

Similarly to Eq. (25), standardized estimation of regression coefficients are equal to

$$\check{\beta}^* = (\check{X}^{*\top} \check{X}^*)^{-1} \check{X}^{*\top} \check{y}^*. \tag{28}$$

On the other hand, the covariance matrix of the estimated regression coefficients is

$$\sigma_{\check{\beta}}^2 = \sigma^2 (X^\top X)^{-1}, \quad \sigma^2 = \sum_{q=1}^{s} (\check{y}_q - \bar{y})^2, \tag{29}$$

where σ^2 is the *error term variance* for X (cf. Definition C.2). After standardization, it is noted that $X^{*\top} X^*$ is just the correlation matrix r_{XX} of X^*. Hence, by Eq. (29) we obtain:

$$\sigma_{\check{\beta}^*}^2 = (\sigma^*)^2 r_{XX}^{-1}. \tag{30}$$

Let $(\text{VIF})_p$ be the p-th diagonal element of the matrix r_{XX}^{-1}. The variance of β_p^* is equal to:

$$\sigma_{\check{\beta}_p^*}^2 = (\sigma^*)^2 (\text{VIF})_p. \tag{31}$$

The diagonal element $(\text{VIF})_p$ is just the variance inflation factor for $\check{\beta}_p^*$. The variance of β_p^* can also be written as [47]

$$\sigma_{\check{\beta}_p^*}^2 = \frac{1}{1 - R_p^2} \Big[\frac{\sigma^2}{\sum_q (x_{qp} - \bar{x}_p)^2} \Big]. \tag{32}$$

With the previous discussion, we conclude that

$$(\text{VIF})_p = \frac{1}{(1 - R_p^2)}, \tag{33}$$

where R_p^2 is defined in C.3.

Theorem C.4. *Let*

$$\textit{VIF} = \frac{\sum_{p=1}^{r} (\textit{VIF})_p}{r - 1}, \tag{34}$$

where r denotes the number of variables in Eq. (13)*. Then* VIF $\geq$ *1 with equality holds if and only if the variables are orthogonal.*

Proof. To prove this, we need to generalize the definition of R^2. By definition,

$$R^2 = \frac{\text{SSE}}{\text{SST}} = \frac{\sum_{q=1}^{s} (\check{y}_q - \bar{y})^2}{\sum_{q=1}^{s} (y_q - \bar{y})^2} = \sum_{q=1}^{s} (\check{y}_q^*)^2. \tag{35}$$

Substituting Eq. (27) into the above identity, we have

$$\sum_{q=1}^{s}(\tilde{y}_q^*)^2 = \sum_{q=1}^{s}(\check{X}_q^*\check{\beta}^*)^2 = (X_q^*\check{\beta}^*)^\top X_q^*\check{\beta}^*, \tag{36}$$

and by Eq. (28), we conclude that

$$R^2 = (r_{XY})^\top (r_{XX})^{-1} r_{XY}. \tag{37}$$

As the finial step, we compute R_p^2 from Definition C.3 by Eq. (37). It should be noted that according to Definition C.3, R_p^2 is the *goodness-of-fit* when X_p is regressed by the r-1 other variables. These variables are uncorrelated in orthonormal case. Hence $r_{XY} = 0, R_p^2 = 0$ and VIF $= 1$. □

Remark. In statistics, when a variable's VIF is greater than 1, or equivalently $R_p^2 \neq 0$, the influence of this variable on the whole estimation is inflated. It breaks the so-called *absence of multicollinearity*, a fundamental principle in multiple regression analysis, and hence causes more error. Since SSE is a constant value, the error generated by the inflation would be counted into SSR, which leads to a decrease in R^2 by Definition C.2 (see [47, 1] for more details).

C.2.3 Statistical Hypothesis Testing

The previous discussion verifies that regressing with orthonormal samples attains a higher *goodness-of-fit*. In other words, it achieves a higher regression accuracy. Tools from *hypothesis testing* are borrowed here to determine the appraisal accuracy mentioned at the beginning of Section C.2.2. We first introduce *mean squared error* (MSE) commonly used in statistics [19]:

Definition C.5. Recall that the residue $e_q = (\hat{y}_q - \check{y}_q)$ from Definition C.2. Then,

$$\text{MSE} = \frac{1}{s}\sum_{q=1}^{s}(\hat{y}_q - \check{y}_q)^2 = \frac{1}{s}\sum_{q=1}^{s}(e_q)^2 = \frac{1}{s}\boldsymbol{e}^\top \boldsymbol{e}. \tag{38}$$

As mentioned in C.2.1, a small coefficient of determination R^2 indicates a large SSR and hence leads to a large MSE. As a result of Theorem C.4, MSE is minimized in the orthonormal case.

We now assume a domain centered at the true value $\boldsymbol{P}_T$ of radius d, and treat the outside space W as the *rejection region*. Statistically, if the distance between $\check{\boldsymbol{P}}$ and $\boldsymbol{P}_T$ is less than a small enough d, we can regard them as the same. Intuitively, if fitted values $\check{\boldsymbol{P}}$ are largely scattered from the true value $\boldsymbol{P}_T$, that is, when MSE is large, it can interfere with our judgment of whether $\boldsymbol{P}$ can be identified with $\boldsymbol{P}_T$. Rigorously, we make a *hypothesis testing* and analyze the probability of rejecting a well-estimated readout function. We prove in the following that when sampling from orthonormal cluster centers, a higher regression accuracy (Theorem C.4) guarantees a lower MSE and therefore increases the appraisal accuracy.

Theorem C.6. *The significance level α_{E_k} reveals that the probability of rejecting a well-estimated readout is lower when sampling from orthonormal centers than sampling from non-orthonormal centers.*

Proof. Let $\boldsymbol{P}$ be a readout function such that $\|\boldsymbol{P}_T - \boldsymbol{P}\| \leq d$ for small enough d. Statistically, we can treat them as the same and simply write $\check{\boldsymbol{P}} = \boldsymbol{P}_T$. In *hypothesis testing*, we define *null hypothesis* H_0 and *alternative hypothesis* H_1 by

$$H_0 : \check{\boldsymbol{P}} = \boldsymbol{P}_T, \quad H_1 : \check{\boldsymbol{P}} \neq \boldsymbol{P}_T, \tag{39}$$

in which H_1 means that we reject a well-estimated readout with H_0 having the opposite meaning. The rejection region for this test is thus given as $W = \{\check{\boldsymbol{P}} \neq \boldsymbol{P}_T\}$. As a conventional procedure in *hypothesis testing*, we take a suitable test statistic $T_{\boldsymbol{E}_k}(\boldsymbol{Z}_i)$ whose distribution f is known [19]. It is used to compute the probability that $\check{\boldsymbol{P}}$ is in the rejection region. The corresponding probability distribution is called potential function $g(\theta)$ for W in this setting:

$$g(\theta) = P_\theta(\check{\boldsymbol{P}} \in W) = \int_W f(T_{\boldsymbol{E}_k}(\boldsymbol{Z}_i))d\boldsymbol{Z}_i \leq \alpha_{\boldsymbol{E}_k}, \quad \theta = H_0 \cup H_1, \tag{40}$$

where the significance level $\alpha_{\boldsymbol{E}_k}$ is the upper bound of the probability of making mistakes (formally called *type I error*) [19].

By Theorem C.4 and Remark C.5, MSE is minimized in the orthonormal case. It can be treated as a variance of distribution f. Then by *Vysochanskij–Petunin inequality*, a refinement of Chebyshev inequality, the integration over W with orthonormal cluster centers $\boldsymbol{E}_k$ is smaller than that with non-orthonormal cluster centers $\boldsymbol{E}'_k$:

$$\int_W f(T_{\boldsymbol{E}_k}(\boldsymbol{Z}_i))d\boldsymbol{Z}_i \leq \int_W f(T_{\boldsymbol{E}'_k}(\boldsymbol{Z}_i))d\boldsymbol{Z}_i. \tag{41}$$

As the result holds true for any well-chosen $T_{\boldsymbol{E}_k}(\boldsymbol{Z}_i)$, $\alpha_{\boldsymbol{E}_k} \leq \alpha_{\boldsymbol{E}'_K}$, this finishes the proof. □

D Running Time

Table 4 shows that state-of-the-art models of Graphormer and SAN are much slower than our BRAINNETTF and VanillaTF, mainly because their implementations are not optimized toward the unique properties of brain networks. Specifically, let e be the number of edges and v be the number of nodes. The calculation of Graphormer and SAN optimizes the case where $e \ll v^2$. However, brain networks usually have a small number of nodes but dense connections, i.e., $e \simeq v^2$. Therefore the optimized sparse graph operations in PyTorch Geometric [23] do not work properly. On the other hand, since the number of nodes in brain networks is usually relatively small (less than 500), we can directly speed up the calculation using matrix multiplication, which is what we did in BRAINNETTF and VanillaTF. Besides, the edge feature generation operator in Graphormer further increases the burden on its computing time.

Table 4: Running time with different graph transformer methods.

Method	Running Time on ABIDE (min)	Running Time on ABCD (min)
SAN	93.01±0.96	908.05±3.6
Graphormer	133.52±0.54	4089.86±5.7
VanillaTF	2.32±0.10	36.26±2.12
BRAINNETTF	**1.98±0.04**	**30.31±1.16**

E Number of Parameters

Table 5: The number of parameters in different models.

Method	#Para on ABIDE	#Para on ABCD
BrainNetCNN	0.93M	0.93M
BrainGB	1.08M	1.49M
FBNetGen	0.55M	1.18M
SAN	57.7M6	186.7M
Graphormer	1.23M	1.66M
VanillaTF	15.6M	32.7M
BRAINNETTF	4.0M	11.2M

F Parameter Tuning

For BrainGB, BrainGNN, FBNetGen, we use the authors' open-source codes. For SAN and Graphormer, we folk their repositories and modified them for the brain network dataset. For BrainNetCNN and VanillaTF, we implement them by ourselves. We use the grid search for some important hyper-parameters for these baselines based on the provided best setting. To be specific, for BrainGB, we search different readout functions {mean, max, concat} with different message-passing functions {Edge weighted, Node edge concat, Node concat}. For BrainGNN, we search different learning rates {0.01, 0.005, 0.001} with different feature dimensions {100, 200}. For FBNetGen, we search different encoders {1D-CNN, GRU} with different hidden dimensions {8, 12, 16}. For

BrainNetCNN, we search different dropout rates {0.3, 0.5, 0.7}. For VanillaTF, we search the number of transformer layers {1, 2, 3} with the number of headers {2, 4, 6}. For SAN, we test LPE hidden dimensions {4, 8, 16}, the number of LPE and GT transformer layers {1, 2} and the number of headers {2, 4} with 50 epochs training. For Graphormer, we test encoder layers {1, 2} and embed dimensions {256, 512}. Furthermore, since the rebuttal time is pretty short, we do not have enough time to dig two new baselines, BrainnetGNN and DGM, which may be why their performance is worse than others.

G Software Version

Table 6: The dependency of BRAINNETTF.

Dependency	Version
python	3.9
cudatoolkit	11.3
torchvision	0.13.1
pytorch	1.12.1
torchaudio	0.12.1
wandb	0.13.1
scikit-learn	1.1.1
pandas	1.4.3
hydra-core	1.2.0

H The Difference between Various Initialization Methods

To show orthonormal initialization can produce more discriminative $\boldsymbol{P}$ between classes than random initialization, we calculate the difference score d based on the formula

$$d = \sum_{i}^{K} \sum_{j}^{V} \frac{|P_{ij}^{female} - P_{ij}^{male}|}{KV}, \tag{42}$$

where V is the number of nodes and K is the number of clusters. After running the t-test, we found the margins between random and orthonormal on both ABIDE and ABCD are significant, which is consistent with our conclusion.

Table 7: The difference score between different initialization methods.

Method	Difference score on ABIDE	Difference score on ABCD
Random	0.067±0.016	0.125±0.010
Orthonormal	0.085±0.015	0.142±0.014