Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Xuan Kan Date

Empower Deep Learning for Brain Network Analysis

By

Xuan Kan
Doctor of Philosophy

Computer Science and Informatics

Carl Yang, Ph.D.
Advisor

Ying Guo, Ph.D.
Co-Advisor

Liang Zhao, Ph.D.
Committee Member

Xiaoxiao Li, Ph.D.
Committee Member

Accepted:

Kimberly Jacob Arriola, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Empower Deep Learning for Brain Network Analysis

By

Xuan Kan
B.E., Tongji University, Shanghai, 2018
M.Sc., Emory University, GA, 2022

Advisor: Carl Yang, Ph.D.
Co-Advisor: Ying Guo, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2024

Abstract

Empower Deep Learning for Brain Network Analysis
By Xuan Kan

Recent large-scale brain network datasets, such as the Philadelphia Neurodevelopmental Cohort (PNC) study and the Adolescent Brain Cognitive Development (ABCD) study, have laid the foundation for applying deep learning techniques to brain network analysis. These datasets provide extensive and diverse brain imaging data and rich phenotypic information, enabling researchers to investigate the complex relationships between brain networks and behavioral measures in large populations. However, applying deep learning to brain network analysis poses several challenges, including the need for better backbone architectures, sample size limitations, and limited supervision signals. This thesis aims to address these challenges by developing innovative deep-learning techniques spanning both model architectures and training strategies.

In the first part of the thesis, we focus on designing novel model architectures tailored for brain network analysis. We propose FBNetGen, an end-to-end differentiable pipeline that generates task-aware functional brain networks from raw fMRI time series data, achieving good performance and providing explainable insights into disorder-specific brain regions and connections. We then introduce Brain Network Transformer (BNT), a transformer-based architecture designed to capture the unique properties of brain networks, demonstrating superior performance on large-scale fMRI datasets. Furthermore, we present Dynamic bRAin Transformer (DRAT), an approach that focuses on modeling dynamic brain networks to capture temporal variations and improve predictions and interpretability.

The second part of the thesis focuses on advanced training strategies to enhance the generalization and performance of deep learning models for brain network analysis. We develop R-mixup, a data augmentation approach operating on the Riemannian manifold of symmetric positive definite matrices, effectively addressing the limited sample size challenge in low-resource settings commonly encountered in neuroimaging studies. Additionally, to obtain richer supervision signals, we propose a multi-task learning framework that jointly predicts various behavioral and clinical measures from brain networks, enabling knowledge sharing across related tasks and improving individual task performance while better utilizing the wide variety of annotated measures available in existing datasets.

Extensive experiments on multiple datasets and tasks demonstrate the superior performance and practical value of our methods. This thesis's contributions facilitate a better understanding of the complex relationships between brain networks and behavioral phenotypes, benefiting neuroimaging research and clinical applications. By addressing the key challenges of better backbone architectures, sample size limitations, and limited supervision signals, this thesis paves the way for more effective and explainable deep learning techniques in brain network analysis, ultimately advancing our understanding of the human brain and its role in cognition and disorders.

Empower Deep Learning for Brain Network Analysis

By

Xuan Kan
B.E., Tongji University, Shanghai, 2018
M.Sc., Emory University, GA, 2022

Advisor: Carl Yang, Ph.D.
Co-Advisor: Ying Guo, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2024

Acknowledgment

I would like to express my deepest gratitude to my advisors, Prof. Carl Yang and Prof. Ying Guo, for their invaluable guidance, support, and encouragement throughout my PhD journey. Their expertise and mentorship have been instrumental in shaping my research and helping me grow as a scholar.

I am forever indebted to my parents for their unwavering support and understanding in my decision to pursue a PhD. Their love and belief in me have been a constant source of motivation and strength.

I want to extend a special thanks to my beloved, Hejie Cui, for her tireless help and being by my side through the ups and downs of this challenging yet rewarding experience. Her love, patience, and encouragement have been my pillars of support.

I am also grateful to Prof. Tianwei Yu for giving me the opportunity to embark on this incredible journey of PhD study. His trust and belief in my potential have been a driving force behind my academic pursuits.

Lastly, I would like to thank all my colleagues, friends, and faculty members who have contributed to my growth and made this experience memorable. Your support and collaboration have been truly invaluable.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 The Importance of Brain Network Analysis

The human brain is a complex system consisting of billions of neurons that are interconnected to form intricate networks. These networks are responsible for various cognitive functions, such as perception, attention, memory, and decision-making. Understanding how these networks are organized and how they give rise to different cognitive processes is a fundamental goal of neuroscience research.

Brain network analysis has emerged as a powerful approach to study the structure and function of the brain. By representing the brain as a graph, with nodes corresponding to brain regions and edges representing the connections between them, we can quantify various properties of brain networks, such as their topology, efficiency, and modularity. These properties can provide insights into how information is processed and integrated across different brain regions, and how these processes are altered in neurological and psychiatric disorders.

One of the key advantages of brain network analysis is that it allows us to study the brain at different scales, from individual neurons to entire brain regions. At the microscopic scale, we can study the connectivity patterns of individual neurons

and how they give rise to local circuits. At the mesoscopic scale, we can study the organization of brain regions into functional modules and how these modules interact with each other. At the macroscopic scale, we can study the global properties of brain networks and how they relate to cognitive and behavioral outcomes, which this dissertation focuses on.

Brain network analysis has been applied to a wide range of neuroimaging modalities, including functional magnetic resonance imaging (fMRI), diffusion tensor imaging (DTI), and electroencephalography (EEG). fMRI, in particular, has become one of the most commonly used modalities for studying brain networks due to its ability to measure blood oxygenation level-dependent (BOLD) signals, which are thought to reflect neuronal activity. By measuring BOLD signals across different brain regions and computing their correlations, we can construct functional brain networks that reflect the statistical dependencies between different brain regions.

Brain network analysis has led to several important discoveries about the organization and function of the brain. For example, studies have shown that the brain exhibits a small-world topology, characterized by high clustering and short path lengths [11]. This topology is thought to be optimal for information processing and integration, as it allows for efficient communication between different brain regions while minimizing wiring costs. Other studies have shown that the brain is organized into functional modules, such as the default mode network and the salience network, which are involved in different cognitive processes [151].

Brain network analysis has also provided important insights into various neurological and psychiatric disorders. For example, studies have shown that patients with Alzheimer's disease exhibit disruptions in functional brain networks, particularly in the default mode network [77]. These disruptions are thought to underlie the cognitive deficits associated with the disease, such as memory loss and difficulties with executive function. Similarly, studies have shown that patients with schizophrenia ex-

hibit alterations in brain network topology, such as reduced clustering and increased path lengths [132]. These alterations are thought to reflect the disconnection between different brain regions that is characteristic of the disorder.

In addition to its clinical applications, brain network analysis has important implications for our understanding of human cognition and behavior. By studying how different brain regions are connected and how these connections change across different cognitive states and tasks, we can gain insights into the neural basis of various cognitive processes, such as perception, attention, and decision-making. For example, studies have shown that the strength of functional connectivity between different brain regions can predict individual differences in cognitive abilities, such as working memory capacity and fluid intelligence [65].

Overall, brain network analysis is a powerful approach for studying the structure and function of the brain, with important applications in both basic and clinical neuroscience research. By representing the brain as a complex network and studying its topological properties, we can gain insights into how the brain processes and integrates information across different scales and how these processes are altered in various neurological and psychiatric disorders. As neuroimaging technologies continue to advance and large-scale brain network datasets become increasingly available, brain network analysis will likely play an even more important role in advancing our understanding of the brain and developing new diagnostic and therapeutic tools for brain disorders.

1.2 The Rise of Deep Learning in Brain Network Analysis

Deep learning has emerged as a powerful tool for analyzing complex, high-dimensional data across various domains, including natural language processing (NLP) and computer vision (CV). In NLP, deep learning models such as BERT [54] and GPT [153]

have revolutionized the field, achieving state-of-the-art performance on tasks such as language translation, sentiment analysis, and question answering. These models can effectively capture the contextual information and semantics of words, enabling them to understand and generate human-like language.

Similarly, in CV, deep learning architectures like convolutional neural networks (CNNs) [123] and ResNet [84] have achieved remarkable success in tasks such as image classification, object detection, and semantic segmentation. These models can automatically learn hierarchical features from raw image data, enabling them to recognize complex patterns and structures.

Inspired by the success of deep learning in NLP and CV, researchers have begun to apply these techniques to brain network analysis [111, 126]. Deep learning methods have the potential to uncover intricate patterns and relationships in complex, high-dimensional brain networks, enabling the discovery of new insights into brain function and disorders. By leveraging the ability of deep learning models to learn hierarchical representations and capture non-linear relationships, we can potentially predict clinical outcomes or classify individuals based on their brain networks with high accuracy.

However, applying deep learning to brain network analysis poses unique challenges compared to NLP and CV. Brain networks are typically represented as graphs, with nodes corresponding to brain regions and edges representing the connections between them. These graphs are often high-dimensional, with the number of edges growing quadratically with the number of nodes. Moreover, brain networks exhibit complex topological properties, such as modularity and small-worldness, which may not be easily captured by traditional deep-learning architectures.

Despite these challenges, there is a growing interest in the neuroimaging community in applying deep learning methods to brain network analysis. Convolutional neural networks, which have been highly successful in CV, have been adapted to

handle graph-structured data, giving rise to graph convolutional networks (GCNs) [118]. GCNs can learn node embeddings by aggregating information from neighboring nodes, enabling them to capture the local and global structure of brain networks. Other deep learning architectures, such as graph attention networks (GATs) [183] and graph transformers [215], have also been proposed to handle the unique properties of brain networks.

In addition to architectural innovations, deep learning models for brain network analysis can benefit from the rich phenotypic information available in large-scale brain network datasets. By incorporating demographic, cognitive, and clinical measures as additional inputs or prediction targets, deep learning models can potentially uncover complex relationships between brain networks and behavior, leading to a more comprehensive understanding of brain function and disorders.

The rise of deep learning in brain network analysis presents an exciting opportunity to advance our understanding of the human brain and develop new tools for diagnosing and treating neurological and psychiatric disorders. By leveraging the power of deep learning and the increasing availability of large-scale brain network datasets, we can potentially uncover new insights into brain organization and function, paving the way for personalized medicine and improved patient outcomes. However, realizing the full potential of deep learning in brain network analysis will require addressing the unique challenges posed by the complexity and high dimensionality of brain networks, as well as the limited sample sizes and supervision signals available in current datasets.

1.3 Large-Scale Brain Network Datasets

Recent large-scale brain network datasets, such as the Philadelphia Neurodevelopmental Cohort (PNC) study [159], the Human Connectome Project (HCP) [181], the

Adolescent Brain Cognitive Development (ABCD) study [22], and the UK Biobank [137], have laid the foundation for applying deep learning techniques to brain network analysis. These datasets provide extensive and diverse brain imaging data along with rich phenotypic information, enabling researchers to investigate the complex relationships between brain networks and behavioral measures in large populations.

1.4 Challenges and Solutions in Applying Deep Learning to Brain Network Analysis

Despite the availability of large-scale datasets, applying deep learning to brain network analysis poses several challenges, including the need for better backbone architectures, sample size limitations, and limited supervision signals.

1.4.1 Better Backbone Architectures

One of the main challenges in applying deep learning to brain network analysis is the need for backbone architectures that can effectively capture the unique properties of brain networks. Traditional deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are not specifically designed to handle the complex, high-dimensional structure of brain networks. To address this challenge, we propose three novel models: FBNetGen [106], Brain Network Transformer (BNT) [107], and Dynamic bRAin Transformer (DRAT) [108].

FBNetGen is an end-to-end differentiable pipeline that generates task-aware functional brain networks from raw fMRI time series data. By jointly optimizing the feature extractor, graph generator, and graph neural network predictor, FBNetGen produces brain networks that are denoised, lower-dimensional, and customized for downstream prediction tasks. This approach achieves good performance and provides explainable insights into disorder-specific brain regions and connections.

BNT is a tailored transformer architecture for modeling the unique properties of brain networks. It leverages effective initial node features based on connection profiles and fully learns the pairwise attention weights to capture the predictive brain network structures. A special readout operator, OCRead, is introduced to fuse node-level embedding to graph-level representations. BNT demonstrates superior performance on large-scale fMRI datasets, such as ABIDE and ABCD.

DRAT focuses on modeling dynamic brain networks for improved predictions and interpretability. It integrates static and dynamic brain networks, being the first attempt to combine multi-view brain networks and fully exploit their complementary information. DRAT incorporates specific attention mechanisms to provide interpretability and insights into the dynamic changes in brain networks.

1.4.2 Sample Size Limitations

Another significant challenge in applying deep learning to brain network analysis is the limited sample size of available datasets. Despite the existence of large-scale brain network datasets, the size of these datasets is still relatively small compared to the high dimensionality of brain networks. This can lead to overfitting when directly applying deep neural networks to brain network data. To tackle this challenge, we develop R-mixup, a data augmentation approach operating on the Riemannian manifold of symmetric positive definite matrices [109].

R-mixup effectively addresses the limited sample size challenge in low-resource settings commonly encountered in neuroimaging studies. By augmenting samples based on Riemannian geodesics, R-mixup preserves the intrinsic geometric structure of the original data and mitigates the swelling effect and arbitrarily incorrect label issues present in existing Mixup methods. Extensive experiments on five biological network datasets spanning both regression and classification tasks demonstrate the effectiveness of R-mixup, especially under low-resource settings.

1.4.3 Limited Supervision Signals

A third challenge in applying deep learning to brain network analysis is the limited supervision signals available in existing datasets. Many brain network datasets only provide a small number of labeled samples for specific tasks, making it difficult to train deep learning models effectively. To overcome this issue, we propose a multi-task learning (MTL) framework that jointly predicts various behavioral and clinical measures from brain networks.

The MTL framework enables knowledge sharing across related tasks and improves individual task performance while better utilizing existing datasets. By training on 35 tasks simultaneously with a shared transformer backbone and task-specific fully connected networks, our approach leverages the diverse range of prediction targets in brain network datasets and improves individual task performance. Moreover, the MTL framework allows us to better leverage existing datasets by utilizing the wide variety of annotated measures available, helping to alleviate the limited sample size challenge. Visualization techniques based on integrated gradients are developed to interpret the learned task correlations and identify salient brain regions and connections, enhancing the interpretability of the predictions.

These innovative solutions to the challenges of better backbone architectures, sample size limitations, and limited supervision signals demonstrate the potential of deep learning techniques in advancing brain network analysis. By addressing these challenges, we can unlock the power of deep learning to uncover complex patterns and relationships in brain networks, leading to a better understanding of brain function and disorders.

Chapter 2

Model Architecture: Task-aware GNN-based fMRI Analysis via Functional Brain NETwork GENeration (FBNetGen)

2.1 Introduction

In recent years, network-oriented analysis has become increasingly important in neuroimaging studies in order to understand human brain organizations in healthy as well as diseased individuals [160, 190, 189, 18, 52]. There are abundant findings in neuroscience research showing that neural circuits are key to understand the differences in brain functioning between populations, and the disruptions in neural circuits largely cause and define brain disorders [99, 192]. Functional magnetic resonance imaging (fMRI) is one of the most commonly used imaging modalities to investigate brain function and organization [73, 130, 168]. There is a strong interest in the neuroimaging community to predict clinical outcomes or classify individuals based on

brain networks derived from fMRI images [111, 199].

Current network analyses typically take the following approach [170, 167, 190]. First, functional brain networks are estimated based on individuals' fMRI data. This is usually done by selecting a brain atlas or a set of nodes/regions of interests (ROI) and extracting fMRI blood-oxygen-level-dependent (BOLD) signal series from each node or brain region. Then, pairwise connectivity is calculated between node pairs using measures such as Pearson correlation and partial correlation. The calculated brain connectivity measures between all the node pairs are then used in the subsequent classification or prediction analyses to classify individuals or predict their clinical outcomes. However, the original BOLD signal series are often high-dimensional and noisy, and the brain networks constructed in this way are not customized towards specific downstream clinical predictions.

There is a growing trend in applying graph neural networks (GNNs) on brain connectivity matrices from fMRI data [200, 4, 125, 126]. GNNs are state-of-the-art deep learning models for graph-structured data which can combine graph structures and node features for various graph-related predictions [118, 195, 184, 218, 201]. However, the mechanism of most GNNs (*i.e.*, message passing) is not compatible with existing functional brain networks which possess both positive and negative weighted edges but no proper node features.

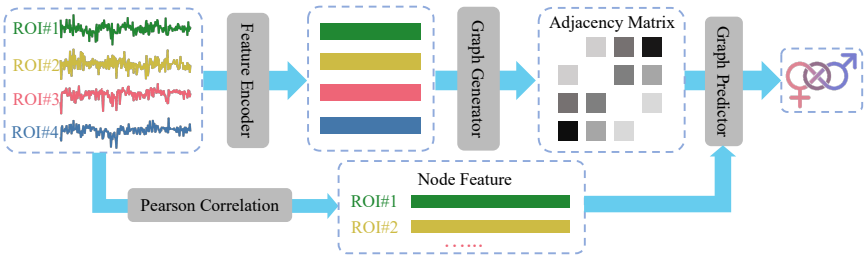


Figure 2.1: Overall framework of our proposed end-to-end task-aware fMRI analysis via functional brain network generation.

In this work, to unleash the power of GNNs in network-based fMRI analysis while

providing valuable interpretability regarding brain region connectivity, we propose to generate functional brain networks that are compatible with GNNs and customized towards downstream clinical predictions from fMRI data. Specifically, we develop an end-to-end differentiable pipeline from BOLD signal series to clinical predictions. Our pipeline includes a feature extractor for denoising and reducing the dimension of raw time-series data, a graph generator for generating individual brain networks from the extracted features, and a graph predictor of GNN for clinical predictions from the generated brain networks (c.f. Figure 2.1).

We conduct extensive experiments using real-world fMRI datasets with the downstream task of gender prediction. FBNetGen achieved consistently better gender prediction accuracy over three types of possible baselines. Furthermore, our in-depth analysis identify a set of brain regions which are useful for predicting gender, which aligns well with existing neurobiological findings.

2.2 Background and Related Work

2.2.1 fMRI-based Brain Network Analysis

fMRI has become the most commonly used imaging modality to probe brain functional organizations by identifying brain functional networks that represent a set of spatially disjoint regions in the brain demonstrating coherent temporal dynamics in fMRI blood oxygen level dependent (BOLD) signals [70]. Functional connectivity (FC) has been found to be related to intrinsic neural processing, cognitive, emotional, visual, and motor functions. Existing studies have shown that FC plays an important role in understanding neurodevelopment, mental disorders and neurodegenerative diseases [68, 61, 132, 41, 173]. There are also findings that reveal gender differences on FC between brain regions [160]. To investigate FC alterations in demographic and clinical subpopulations, the commonly adopted methods include edge-wise tests for

between group differences [144, 31, 117], tests for detecting coordinated disruptions across multiple brain subsystems [216, 88], graph theory based methods for comparing brain network graph metrics [157, 156, 67] and graphical model approaches [135, 87, 121] .

2.2.2 Graph Neural Networks

Graph Neural Networks (GNNs) have revolutionized the field for modeling various important real-world data in the form of graphs or networks [118, 81, 183, 28, 165], such as social network, knowledge graphs, protein-interaction networks, etc.The advantage of GNNs is that they can combine node features and graph structures in an end-to-end fashion towards downstream prediction tasks. Various delicately designed GNN models have been developed for graph classification. For example, GCN [118] is one of the basic and most representative GNNs that generalized the shared filter for graphs from the successful CNNs models in computer vision; Velickovic et al. [183] integrated the attention mechanism to assign different weights for neighbors in each graph convolution layers; Xu et al. [195] proposed graph isomorphism network, which is a simple yet powerful architecture that is proved to have equal discriminative ability with the 1-WL test.

Yet, until recently, some emerging attention has been devoted to the generalization of GNN-based models to fMRI-based brain network analysis [125, 126]. However, GNNs require explicitly given graph structures and node features, which are typically not available in brain networks and usually constructed manually based on statistical correlations [170]. In addition, only one recent study has considered the learnable generation of brain networks but without downstream tasks [224], and no study has explored the interpretability of the generated brain networks towards downstream tasks, which is critical in neuroimaging research regarding its practical scope and promising social impact.

2.3 FBNetGen

2.3.1 Overview

In this section, we elaborate the design of FBNetGen and its three main components as shown in Figure 2.1. Specifically, the input $\boldsymbol{X} \in \mathbb{R}^{n \times v \times t}$ denotes the BOLD time-series for regions of interest (ROIs) as the input, where n is the sample size, v is the number of ROIs and t is the length of time-series. Each $\boldsymbol{x} \in \mathbb{R}^{v \times t}$ represents a sample (individual). The target output is the prediction label $\boldsymbol{Y} \in \mathbb{R}^{n \times |\mathcal{C}|}$, where $\mathcal{C}$ is the class set of Y and $|\mathcal{C}|$ is the number of classes. As an intermediate output of the end-to-end pipeline, we also generate a functional brain network $\boldsymbol{A} \in \mathbb{R}^{v \times v}$ (*i.e.*, brain connectivity matrix between ROIs) for each sample $\boldsymbol{x} \in \mathbb{R}^{v \times t}$ through graph generator, which highlights prediction-specific prominent brain network connections and provides unique interpretation towards neuroscience research.

2.3.2 Feature Encoder

BOLD signal series is a type of temporal sequence data. The main difference between BOLD signal series with ordinary time-series data is that BOLD is a group of aligned sequences instead of independent ones. Traditional BOLD analysis methods like ICA [180] and PCA [177] ignore the temporal order in BOLD, which means shuffling the time steps does not change the dimension reduction results. Besides, PCA and ICA can only capture linear information within or across time-series. Finally, to construct brain networks from ICA and PCA results, Pearson correlation is often adopted [170], but the brain networks generated in this way are not aware of the downstream tasks and not compatible with GNNs (due to containing negative edge weights).

Recently, deep neural network has shown great success in capturing complex non-linear information on various time-sequence tasks, such as natural language process, market analysis and traffic control [163, 36, 119]. Therefore, we apply two commonly

used deep encoding models for time-series data, 1D-CNN and bi-GRU, for feature extraction from BOLD signals. Specially, we choose bi-GRU rather than LSTM since bi-GRU can achieve similar performance with less parameters [36]. This is vital for brain network analysis because the sample size of fMRI dataset is usually pretty small (*e.g.*, less than 1000), and a lightweight framework is essential to avoid over-fitting.

Specifically, when the feature encoder is set as u-layer 1D-CNN, the process of generating node features $\boldsymbol{h}_e \in \mathbb{R}^{v \times d}$ for v ROIs can be decomposed as

$$\boldsymbol{h}^u = \mathrm{CONV}_u(\boldsymbol{h}^{u-1}), \tag{2.1}$$

$$\boldsymbol{h}_e = \mathrm{MLP}(\mathrm{MAXPOOL}(\boldsymbol{h}^u)), \boldsymbol{h}_e \in \mathbb{R}^{v \times d}, \tag{2.2}$$

where $\boldsymbol{h}^0 = \boldsymbol{x}$ is the original BOLD signal sequence, d is the embedding size of each ROI and the kernal size of CONV_1 equals the window size τ. Similarly, when the feature encoder is set as bi-GRU, the process can be decomposed as

$$\boldsymbol{h}_r = \mathrm{biGRU}([\boldsymbol{x}^{(z\tau-\tau):z\tau}]), \boldsymbol{h}_r \in \mathbb{R}^{v \times 2\tau}, \tag{2.3}$$

$$z = 1, \cdots, \lfloor \frac{t}{\tau} \rfloor, \tag{2.4}$$

where $[\boldsymbol{x}^{(z\tau-\tau):z\tau}]$ represents splitting the input sequence x into z segments of length τ. Finally, a MLP layer is applied to generate the final embedding of size d for each ROI

$$\boldsymbol{h}_e = \mathrm{MLP}(\boldsymbol{h}_r), \boldsymbol{h}_e \in \mathbb{R}^{v \times d}. \tag{2.5}$$

2.3.3 Graph Generator

Between encoder and predictor, a feature-based and task-oriented functional brain network is generated from $\boldsymbol{h}_e$. It is formulated as the connectivity matrix $\boldsymbol{A}$, which stores the pair-wise connectivity strengths between ROIs as elements. Unlike the

commonly used traditional approach for functional brain network construction that calculates the pairwise Pearson correlations between raw time-series of ROIs [170], we generate a learnable $\boldsymbol{A}$ based on encoded time-series features as

$$\boldsymbol{h}_A = \text{softmax}(\boldsymbol{h}_e), \tag{2.6}$$

$$\boldsymbol{A} = \boldsymbol{h}_A \boldsymbol{h}_A^T, \tag{2.7}$$

which can be regularized by the downstream prediction task through an end-to-end framework. The softmax operation highlights the strong ROI connections by generating skewed positive edge weights, which are compatible with GNNs and valuable for interpretation. In contrast, brain connectivity matrices generated from traditional (*i.e.*, statistical) methods are not compatible with GCN, since they contain negative weights (correlation scores can be negative) in edges.

Considering the limited supervision in neuroscience research, sheer supervision is usually not enough to fit a model very well. For instance, when the windows size τ is set as 8, the parameters number of 1D-CNN encoder can reach up to 20k, while there are only 353 samples in PNC training set. It becomes even harder for the model to generate high-quality graphs when the gradient feedback is too long.

In order to facilitate the learning of brain networks beyond the sheer supervision of graph-based classification, we consider incorporating exterior regularizations that are in line with previous scientific studies. It is shown in literature [179, 160] that there are edge-level difference between genders in both structural and functional brain connectivity matrices. Based on these observations, we further apply three group-based and structure-based regularizers during training, named as group intra loss, group inter loss and sparsity loss, respectively.

Group Intra Loss. Previous clinical findings [172] show that there are consistent patterns among individuals in resting-state functional connectivity. In order to utilize

the latent consistent patterns as a regularization for our model training, we design a group intra loss, which aims to minimize the difference of connectivity matrices within a class.

Given a class $c \in \mathcal{C}$, $\mathcal{S}^c = \{i \mid Y_{i,c} = 1\}$ is the set containing all samples' index whose label is c. With the mean μ_c and variance σ_c^2 of all samples' $\boldsymbol{A}$ with label c within a batch computed as

$$\mu_c = \sum_{k \in \mathcal{S}^c} \frac{\boldsymbol{A}^k}{|\mathcal{S}^c|},\ \sigma_c^2 = \sum_{k \in \mathcal{S}^c} \frac{\left\|\boldsymbol{A}^k - \mu_c\right\|_2^2}{|\mathcal{S}^c|}, \tag{2.8}$$

the group intra loss can be effectively calculated in $\mathcal{O}(n)$ time as

$$L_{intra} = \sum_{c \in \mathcal{C}} \sum_{i \in \mathcal{S}^c} \frac{\|\boldsymbol{A}^i - \mu_c\|_2^2}{|\mathcal{S}^c|} = \sum_{c \in \mathcal{C}} \sigma_c^2. \tag{2.9}$$

Group Inter Loss. Cognition science findings in [160] substantiate that there are significant difference among the functional brain networks of different genders, such as brain volume. Hence, we incorporate a group inter loss that aims to maximize the difference of connectivity matrices across different classes, while keeping those within the same class similar. With proper derivation, this loss can also be calculated in $\mathcal{O}(n)$ time as

$$\begin{aligned} L_{inter} &= \sum_{a,b \in \mathcal{C}} (\sigma_a^2 + \sigma_b^2 - \frac{\sum_{i \in \mathcal{S}^a} \sum_{j \in \mathcal{S}^b} \|\boldsymbol{A}^i - \boldsymbol{A}^j\|_2^2}{|\mathcal{S}^a||\mathcal{S}^b|}) \\ &= -\sum_{a,b \in \mathcal{C}} \|\mu_a - \mu_b\|_2^2 . \end{aligned}$$

Sparsity Loss. The model with only group loss may overemphasize the graph difference between genders, which could harm the model's performance and stabilization. To mitigate the degree of deviation caused by large values in the generated graph and highlight the most contributory task-specific ROI connections, we further enforce the

sparsity of generated brain networks, where a sparsity loss is formulated as

$$L_{sparsity} = \frac{1}{nvv} \sum_{i=1}^{n} \left\| \text{vec}(\boldsymbol{A}^i) \right\|_1 . \tag{2.10}$$

2.3.4 Graph Predictor

We apply GNN on the constructed graphs for prediction. GNN is a powerful tool that can learn node representations by transforming, propagating and aggregating node features and graph structure information.

In practice, we initialized node feature $\boldsymbol{F}_p$ for the node p as a vector of Pearson correlation scores between its time-series data with those of all nodes contained in the graph. With the initial node features $\boldsymbol{F} \in \mathbb{R}^{v \times f}$ of ROIs and the learnable connectivity matrix $\boldsymbol{A}$ from graph generator, we can apply a k-layer graph convolutional network [118] to update the node embeddings through

$$\boldsymbol{h}^k = \text{ReLU}\left(\boldsymbol{A}\boldsymbol{h}^{k-1}W^k\right), \tag{2.11}$$

where W^k represents learnable parameters in convolutional layers and $\boldsymbol{h}^0 = \boldsymbol{F}$. The graph-level embedding is obtained by summing up all the node embeddings after the final convolutional layer. A BatchNorm1D is then applied to avoid extreme large values. Finally, another MLP function is employed for classification prediction,

$$\hat{y} = \text{MLP}\left(\text{BatchNorm1D}\left(\sum_{p=1}^{v} \boldsymbol{h}_p^k\right)\right). \tag{2.12}$$

2.3.5 End-to-end Training

We combine the aforementioned three components into end-to-end training, where the label information y and the task-oriented graphs are leveraged at the same time. Another advantage of end-to-end training is that the feature encoder provides a larger

parameter search space than a pure GNN prediction model, leading to potential performance improvements. Furthermore, the intermediate graphs guided by prediction tasks are generated as a by-product, providing explicit task-oriented explanation for deep model's prediction. Overall, our final training objective is composed of four terms:

$$L = L_{ce} + \alpha L_{intra} + \beta L_{inter} + \gamma L_{sparsity}, \tag{2.13}$$

where L_{ce} is the supervised cross-entropy loss for prediction, and α, β, γ are tunable hyper-parameters representing the weights of three regularizers.

2.4 Experiments

In this section, we evaluate the effectiveness and interpretability of FBNetGen with extensive experiments. For effectiveness, we aim to answer the following research questions.

- **RQ1.** How does FBNetGen perform compared with possible baselines? Specifically, we compare with models of three types including (a) directly using time-series features without graph construction; (b) using traditional (statistical) methods to construct graphs instead of learnable generation; (c) using other learnable graph generators.
- **RQ2.** How do different components in our graph generator affect the performance of FBNetGen?
- **RQ3.** How do the hyper-parameters influence FBNetGen and the compared models' performance?

For interpretability, we aim to investigate the advantages of FBNetGen regarding the consistency between its learned important brain network connectivity patterns and existing neuroscience discoveries.

Table 2.1: Performance comparison with three types of baselines.

Type	Method	Dataset: PNC		Dataset: ABCD	
		AUROC	Accuracy	AUROC	Accuracy
Time-series	1D-CNN	63.7 ± 3.8	54.7 ± 1.2	68.1 ± 3.1	63.2 ± 2.6
	bi-GRU	65.1 ± 3.5	58.1 ± 2.4	51.2 ± 1.0	49.9 ± 0.8
Traditional Graph	GNN-Uniform	70.6 ± 4.8	66.2 ± 3.9	88.8 ± 0.7	80.5 ± 0.7
	GNN-Pearson	69.6 ± 4.5	65.6 ± 3.3	88.8 ± 0.3	80.7 ± 0.6
Learnable Graph	LDS-GRU	75.4 ± 3.2	69.2 ± 5.8	89.6 ± 0.5	80.4 ± 1.6
	LDS-CNN	75.7 ± 3.8	69.8 ± 6.2	90.0 ± 0.3	82.5 ± 0.9
	GTS	68.2 ± 1.9	63.7 ± 2.4	87.8 ± 1.1	76.7 ± 2.0
Ours	FBNETGNN-CNN	79.4 ± 2.8	70.4 ± 2.3	91.5 ± 0.5	**82.5 ± 0.7**
	FBNETGNN-GRU	**82.7 ± 4.7**	**77.7 ± 3.9**	**91.6 ± 0.4**	81.6 ± 1.7

2.4.1 Experimental Settings

Dataset. We conduct experiments demonstrating the utility of FBNetGen using two real-world fMRI datasets.

The first dataset is from the Philadelphia Neuroimaging Cohort (PNC), a collaborative project from the Brain Behavior Laboratory at the University of Pennsylvania and the Children's Hospital of Philadelphia. It includes a population-based sample of individuals aged 8–21 years [159]. After excluding subjects with excessive motion [160, 190], 503 subjects' rs-fMRI data were included in our analysis. Among these subjects, 289 (57.46%) are female, which indicates that our dataset is balanced across genders. In our paper, we adapt the 264-node atlas defined by [152] for connectivity analysis. The nodes are grouped into 10 functional modules that correspond to major resting state networks [169]. Standard pre-processing procedures are applied to the rs-fMRI data. For rs-fMRI, the pre-processing include despiking, slice timing correction, motion correction, registration to MNI 2mm standard space, normalization to percent signal change, removal of linear trend, regressing out CSF, WM, and 6 movement parameters, bandpass filtering (0.009–0.08), and spatial smoothing with a 6mm FWHM Gaussian kernel. In the resulting data, each sample contains 264 nodes with time series data collected through 120 time steps. For connectivity analysis, we

focus on the 232 nodes in the Power's atlas that are associated with major resting state functional modules [169].

The second dataset from Adolescent Brain Cognitive Development Study (ABCD) [22], is one of the largest publicly available fMRI datasets. This study is recruiting children aged 9–10 years across 21 sites in the U.S. Each child is followed into early adulthood, with repeated imaging scans as well as extensive psychological and cognitive testing. This study started in 2016 and releases data in regular intervals. We use rs-fMRI scans for the baseline visit processed with the standard and open-source ABCD-HCP BIDS fMRI Pipeline [1]. The HCP 360 ROI atlas template is used for each subject's data [74]. After processing, each sample contains a connectivity matrix whose size is 360×360 and BOLD time-series for each node. Since different sample's BOLD time-series have different lengths, only samples with at least 512 time points are selected, and only the first 512 time points for each sample are included in the subsequent analysis. After this selection, 7901 children were included in the analysis. Among them, 3961 (50.1%) are female and 3940 (49.9%) are male. For interpretability analysis, we organize nodes into communities using the AAc-6 parcellation scheme provided by [3], which divides the 360 ROIs into 6 functional modules.

Metrics. We choose gender prediction as the evaluation task, with available labels from both PNC and ABCD datasets. Since gender prediction is a binary classification problem and both PNC and ABCD datasets are balanced across classes, AUROC is the most comprehensive performance metric and is adopted here for fair performance comparison. Besides, we also include accuracy as a metric to reflect the practical prediction performance of FBNetGen.

Implementation details. For experiments on the two different feature encoders, we restrict the number of 1D-CNN layer u as 3 since the dataset is relatively small. The detailed design of the 1D-CNN encoder can be found in Appendix A.1. Regarding

[1] https://github.com/DCAN-Labs/abcd-hcp-pipeline

GRU feature encoder, we set the number of layers to 4. For both feature encoders, the embedding size d of $\boldsymbol{h}_e$ is searched from $\{4, 8, 12\}$, and the window size τ is a tuned from different ranges based on the sequence length of each dataset, with $\{4, 6, 8\}$ for PNC, and $\{8, 16, 32\}$ for ABCD. For the graph generator, the weights of each loss component α, β, γ are set as 10^{-3}, 10^{-3} and 10^{-4}, respectively. As for the graph predictor, the number of GCN layers is set as 3 following the common practice. We randomly split 70% of the datasets for training, 10% for validation, and the remained are utilized as the test set. In the training process of FBNetGen, we use the Adam optimizer with an initial learning rate of 10^{-4} and a weight decay of 10^{-4}. The batch size is set as 16. All the models are trained for 500 epoches and that achieves the highest AUROC performance on the validation set is tested for performance comparison. All the reported performances are the average results of 5 runs. Please refer to the supplementary material for all code of the implementation of FBNetGen.

Computation complexity. In FBNetGen, the computation complexity of feature encoder, graph generator and graph predictor are $\mathcal{O}(\mu v t)$, $\mathcal{O}(v^2)$ and $\mathcal{O}(k v^2)$ respectively, where μ is the layer number of feature encoder, v is the number of ROIs, t is the length of time-series, and k is the layer number of graph predictor. The overall computation complexity of FBNetGen is thus $\mathcal{O}(v(v + t))$.

Table 2.2: AUROC performance with different regularizers

Dataset	**PNC**				**ABCD**			
Regularizers	All	CE	CE+GL	CE+SL	All	CE	CE+GL	CE+SL
FBNETGNN-CNN	**79.4 ± 2.8**	75.4 ± 2.7	76.4 ± 3.5	77.0 ± 1.8	**91.5 ± 0.5**	91.0 ± 0.8	**91.5 ± 0.4**	91.3 ± 0.9
FBNETGNN-GRU	**82.7 ± 4.7**	75.6 ± 1.4	76.2 ± 3.8	78.8 ± 1.7	**91.6 ± 0.4**	91.3 ± 0.5	91.3 ± 0.4	91.5 ± 0.7

2.4.2 RQ1: Performance Comparison

We compare FBNetGen with baselines of three types.

(a) FBNetGen vs. models directly using time-series features.

We compare our graph-based model with two baselines that directly model time-series data without graph construction. Two commonly used models for temporal sequence, 1D-CNN and bi-GRU, are applied to encode BOLD time-series, which are the same architectures as the feature encoder of FBNetGen. In both time-series baselines, the feature encoder is directly followed by a multilayer perceptron that makes predictions based on the encoded features without building brain networks. The performance of these two baselines is presented in the 'Time-series' columns in Table 2.1. To ensure fairness, all hyper-parameters are shared across time-series baselines and the feature encoder of our model. It can be seen that our FBNetGen with the same feature encoders outperform their corresponding time-series baselines by significant margins (up to 20% absolute improvements). This suggests the necessity of intermediate brain network generation for effective brain network analysis, which is consistent with recent understanding in neural science [99, 192].

(b) FBNetGen vs. models using traditional methods to construct graphs.

To investigate the advantage of our task-oriented learnable graph generator over traditional statistics-based methods, we compare our task-oriented learnable graphs $\boldsymbol{A}$ calculated from Eq. (2.7) with two widely practiced traditional ways to construct brain networks. One of the most popular methods to construct brain networks is the Pearson graphs [170]. Specifically, in the constructed brain network $\mathbf{A}^P$ of Pearson graphs, each entry is calculated as the absolute value of Pearson correlation coefficient between two raw time-series

$$\mathbf{A}^P_{p,q} = |\text{cov}(\boldsymbol{x}_p, \boldsymbol{x}_q)|, \tag{2.14}$$

where cov represents the co-variance function. To isolate the influence of node features, we also use the uniform graphs $\mathbf{A}^U$ as a control group, which corresponds to setting the adjacency matrixes with all 1's

$$\mathbf{A}^U = \{1\}^{v \times v}. \tag{2.15}$$

All these two types of graphs are paired with the same node features $\boldsymbol{F}$ and then processed by the same GNN graph predictor as FBNetGen for downstream prediction. As shown in the 'Tradition Graph' columns in Table 2.1, GNN with our learnable graph gains prominent improvements (up to 13% absolute improvements) compared with the traditional graphs, indicating that our task-oriented learnable graphs are more informative and compatible with GNNs.

(c) FBNetGen vs. models using other learnable graph generators.

We further introduce another three baselines based on learnable graph generators, namely LDS-GRU, LDS-CNN and GTS. LDS [69] is a framework which joint learns graph structures and model parameters through bilevel optimization when graph structures are not available. However, the graph generator setting of LDS is different from our framework. Specifically, LDS targets at learning a discrete population graph which contains all samples as nodes, whereas in our setting, we learn a weighted graph as the brain network for each individual sample. To ensure a fair comparison, we adapt our two feature encoders, 1D-CNN and GRU, to the bilevel optimization framework of LDS, and call them LDS-GRU and LDS-CNN, respectively. All hyper-parameters of LDS-GRU and LDS-CNN are set to the same as their corresponding feature encoders in FBNetGen. Another existing method that can learn graph structure from a group of time-series data is GTS [163], which combines time-series data and graph structure to make sequence-to-sequence prediction. Here GTS is revised to generate

classification result for each sample to suit our gender prediction task. As we can observe from the 'Learnable Graph' type in Table 2.1, our task-oriented graph generator FBNetGen-GRU and FBNetGen-CNN consistently overperform all three baselines using other graph generators, which demonstrate the superior advantage of our proposed end-to-end framework tailored for fMRI analysis.

2.4.3 RQ2: Ablation Studies

We further examine the major designs in in our graph generator (Section 2.3.3): the Group Loss (GL), including both group inner loss and group intra loss, and the Sparsity Loss (SL). We vary the original model with the Cross Entropy Loss (CE), GL and SL by removing each component once at a time, and observe the performance of each ablated model variant. The results are shown in Table 2.2. For training curves, please refer to Figure A.1 in Appendix A.2. From the training curves and the final performance, we see that on PNC dataset, the original model with all designed components improves more stably than its three ablated versions and finally achieves the highest performance. Specifically, CE+SL achieves close-to-optimal performance, demonstrating the effectiveness of sparsity regularizers in generating informative brain networks. On the ABCD dataset, we observe similar trends among the model variants but slightly smaller gains brought by the GL and SL regularizers, because their effectiveness is more significant when the training data are limited (such as in PNC).

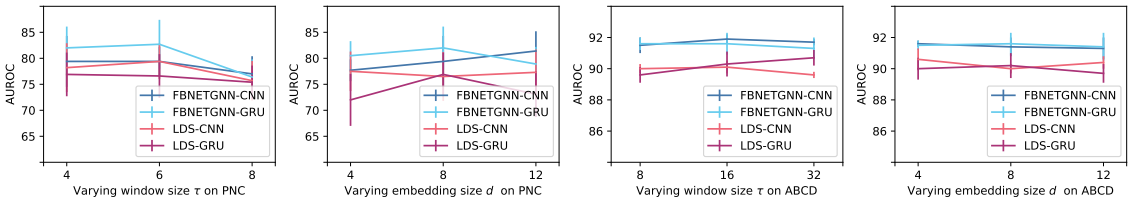


Figure 2.2: Influence of two hyper-parameters in the feature encoder of FBNetGen and LDS based baselines.

2.4.4 RQ3: Influence of Hyper-parameters

We investigate two hyper-parameters that are most influential to the performance of the compared models, namely window size τ and embedding size d in the feature encoders of 1D-CNN and GRU. To reflect the influence comprehensively, we also include the LDS baselines that also use these feature encoders. The results of adjusting hyper-parameters on PNC and ABCD datasets are shown in Figure 2.2. As we can observe from Figure 2.2 (a), window size should not be overly large since the BOLD sequence of PNC dataset is relatively short. Also, increasing the embedding size does not necessarily improve the overall performance of FBNetGen. Shifting perspective to the larger dataset ABCD, we find that the values of both hyper-parameters do not influence much on it than on the smaller dataset of PNC, demonstrating the stable performance of FBNetGen when training data are more sufficient. It is impressive that our FBNetGen consistently achieves better performance compared with the baselines of LDS-CNN and LDS-GRU, in the large ranges of hyper-parameters. This highlights the reliable supremacy of FBNetGen over other graph generators.

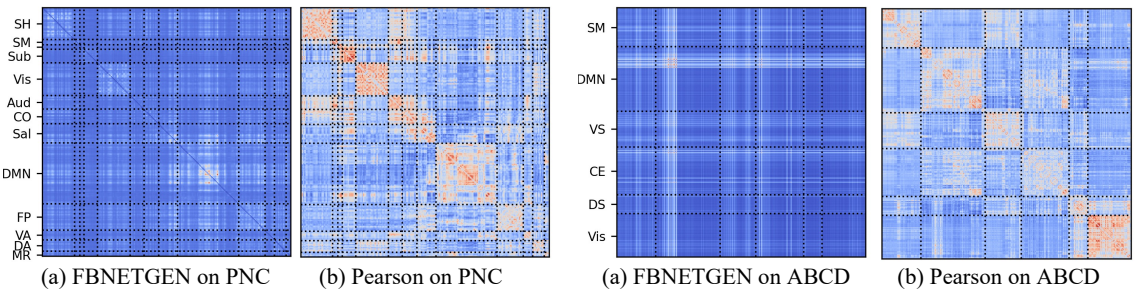


Figure 2.3: Visualizations of learnable graph vs. Pearson graph. Warmer colors indicate higher values. Abbreviations of neural systems: On PNC, SH (Somatomotor Hand), SM (Somatomotor Mouth), Sub (Subcortical), Vis (Visual), Aud (Auditory), CO (Cingulo-opercular), Sal (Salience), DMN (Default mode), FP (Fronto-parietal), VA (Ventral attention), DA (Dorsal attention), MR (Memory retrieval), On ABCD, SM(Somatomotor), DMN (Default mode), VS (Ventral salience), CE (Central executive), DS (Dorsal salience), Vis (Visual).

2.4.5 Interpretability Analysis

In this section, we visualize and compare our learnable graphs with the most commonly used existing functional brain networks, which are functional connectivity based on Pearson correlation [170]. Our results indicate that our learnable graph approach is more task-oriented and advantageous in capturing differences among classes.

We use the average graph across all samples to demonstrate the predominant neural systems across subjects. The mean heatmap visualizations of our learnable brain graphs and the Pearson brain graphs are shown in Figure 2.3. As is shown by the heat values, our graph distinctively and consistently highlights the default mode network (DMN) for both the PNC and ABCD datasets. This aligns with previous neurobiological findings using the PNC data [160], which identify regions with significant differences between genders within the DMN. This consistency remains across both datasets and both brain atlases, validating that our method yields reliable results reproducible across studies. Furthermore, the interpretation of task-specific brain regions and connections are potentially useful for the analysis of other clinical prediction tasks where disease-region relevance is unclear. In contrast, in the Pearson graph the most significant positive components are the connections within functional modules. These within-module connections reflect intrinsic brain functional organizations but are not necessarily informative for predicting gender.

To demonstrate that our learnable graphs possess discrimination ability among classes, we divide these learnable graphs based on genders, and apply t-tests to identify edges $\mathcal{E}^d$ with significantly different strengths ($p < 0.05$) between genders. A difference score T is designed to reflect the discrimination ability. The difference score T_u of each predefined functional module u is calculated as

$$T_u = \sum_{(p,q)\in\mathcal{E}^d} \frac{\mathbb{1}(p \in \mathcal{M}_u) + \mathbb{1}(q \in \mathcal{M}_u)}{2v|\mathcal{M}_u|}, \tag{2.16}$$

where v is the number of ROIs and $\mathcal{M}_u$ is a set containing all indexes of nodes belonging to the module u. Higher scores indicate larger differences between genders. For the PNC data, the top 3 functional modules are memory retrieval network, default mode network and ventral attention network. For the ABCD data, the top 2 modules are default mode network and ventral salience network (including ventral attention network). Literature [160] indicates that ROIs with significant sex differences are located within the default mode network, the ventral attention network, the auditory network, and the memory retrieval network, which aligns well with our top ranked modules; whereas Pearson graphs cannot match the literature as well as our learnable graphs. For more details, please refer to Appendix A.3. This observation further validates that our learnable graphs can effectively capture group differences in brain network and facilitate fMRI-based brain network classification.

2.5 Conclusion

In this paper, we present FBNetGen, a task-aware GNN-based framework for fMRI analysis via functional brain network generation, which generates the brain connectivity matrices and predicts clinical outcomes simultaneously from fMRI BOLD signal series. Extensive experiments demonstrate that FBNetGen consistently outperforms three types of possible baselines including directly using time-series features without graph construction; using traditional methods to construct graphs; and using other learnable graph generators. Besides, the interpretation analysis of our learnable brain networks shows the generated networks are task-oriented and possess the discrimination ability among classes, providing aligned interpretation towards neurobiological findings. Our framework is immediately usable in practice for exploring more real-world prediction tasks such as mental disease diagnosis and mental disorder analysis. In the future, we plan to test our techniques on more datasets and tasks,

improve the graph generator beyond direct link prediction, as well as apply pretraining and transfer learning techniques to learn commonly important brain connectivity structures across multiple datasets and tasks.

Chapter 3

Model Architecture: Brain Network Transformer

3.1 Introduction

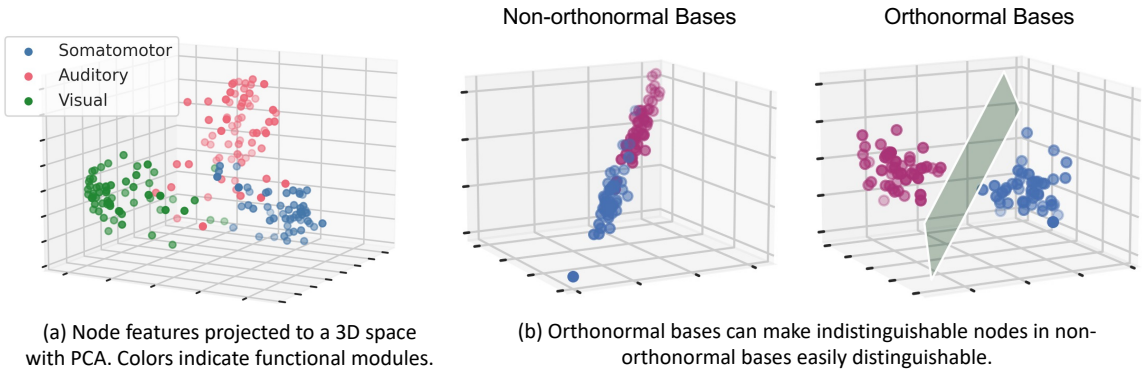
Brain network analysis has been an intriguing pursuit for neuroscientists to understand human brain organizations and predict clinical outcomes [160, 190, 189, 18, 52, 80, 166, 87, 189, 88, 122, 135, 90]. Among various neuroimaging modalities, functional Magnetic Resonance Imaging (fMRI) is one of the 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 [170, 167, 190, 49]. 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 [182, 51] and

computer vision [56, 35, 178]. Recent efforts have also emerged to apply Transformer-based designs to graph representation learning. GAT [183] firstly adapts the attention mechanism to graph neural networks (GNNs) but only considers the local structures of neighboring nodes. Graph Transformer [59] injects edge information into the attention mechanism and leverages the eigenvectors of each node as positional embeddings. SAN [120] 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. Graphomer [207], which achieves the first place on the quantum prediction track of OGB Large-Scale Challenge [89], 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 [126, 106, 46, 202, 226]. 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 [45, 78]. 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 [46]. 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 3.1: 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 [46], 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 [106]. 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 [111, 126, 46]. As is shown in Figure 3.1(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 [20]. 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 [194] 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 3.1(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 [19] 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 [22] show significant improvements brought by our proposed Brain Network Transformer.

3.2 Background and Related Work

3.2.1 GNNs for Brain Network Analysis

Recently, emerging attention has been devoted to the generalization of GNN-based models to brain network analysis [125, 2]. GroupINN [200] utilizes a grouping-based layer to provide explainability and reduce the model size. BrainGNN [126] 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 [44] proposes an interpretable framework to analyze disorder-specific ROIs and prominent connections. In addition, FBNetGen [106] considers the learnable generation of brain networks and explores the explainability of the generated brain networks towards downstream tasks. Another benchmark paper [46] systematically studies the effectiveness of various GNN designs over brain network data. Different from other work focusing on static brain networks, STAGIN [114] utilizes GNNs with spatio-temporal attention to model dynamic brain networks extracted from fMRI data.

3.2.2 Graph Transformer

Graph Transformer raises many researchers' interest currently due to its outstanding performance in graph representation learning. Graph Transformer [59] firstly injects edge information into the attention mechanism and leverages the eigenvectors as positional embeddings. SAN [120] enhances the positional embeddings and improves the attention mechanism by emphasizing neighbor nodes while incorporating the global information. Graphomer [207] designs unique mechanisms for molecule graphs and achieves the SOTA performance. Besides, a fine-grained attention mechanism is de-

veloped for node classification [225]. Also, the Transformer is extended to larger-scale heterogeneous graphs with a particular sampling algorithm in HGT [91]. EGT [96] further employs edge augmentation to assist global self-attention. In addition, LSPE [60] leverages the learnable structural and positional encoding to improve GNNs' representation power, and GRPE [148] enhances the design of encoding node relative position information in Transformer.

3.3 Brain Network Transformer

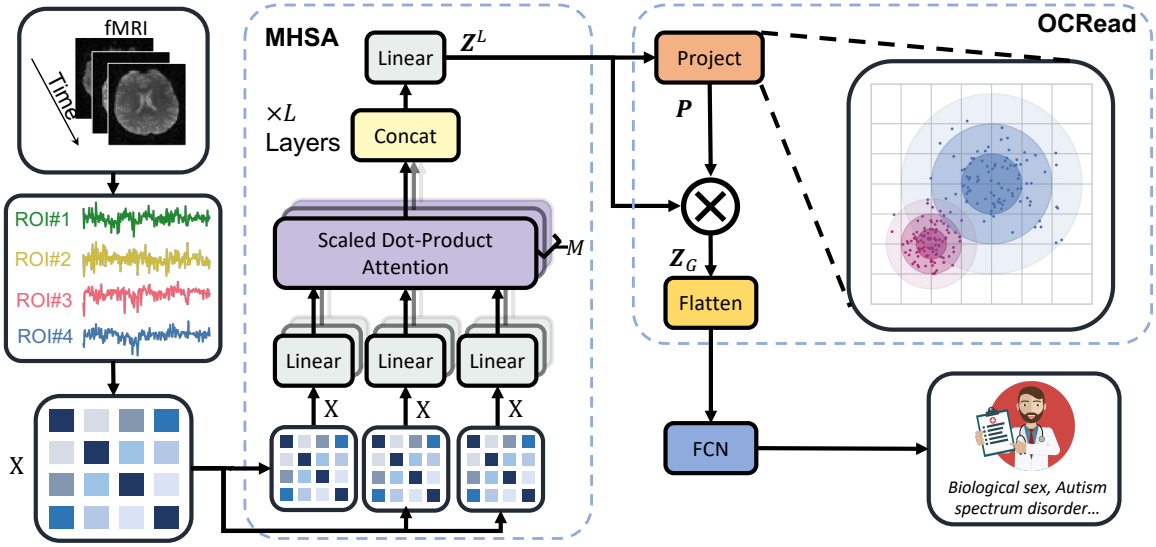


Figure 3.2: The overall framework of our proposed Brain Network Transformer.

3.3.1 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 3.2, 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.

3.3.2 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 [126, 106, 46]. 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.2). As for the attention mechanism, previous work [46] 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 $\boldsymbol{Z}^L = \text{MHSA}(\boldsymbol{X}) \in \mathbb{R}^{V \times V}$. For each layer l, the output $\boldsymbol{Z}^l$ is obtained by

$$\boldsymbol{Z}^l = (\|_{m=1}^{M} \boldsymbol{h}^{l,m}) \boldsymbol{W}_{\mathcal{O}}^l, \boldsymbol{h}^{l,m} = \text{Softmax}\left(\frac{\boldsymbol{W}_{\mathcal{Q}}^{l,m} \boldsymbol{Z}^{l-1} (\boldsymbol{W}_{\mathcal{K}}^{l,m} \boldsymbol{Z}^{l-1})^\top}{\sqrt{d_{\mathcal{K}}^{l,m}}} \right) \boldsymbol{W}_{\mathcal{V}}^{l,m} \boldsymbol{Z}^{l-1}, \quad (3.1)$$

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 $\boldsymbol{W}_{\mathcal{K}}^{l,m}$.

3.3.3 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. $\text{Mean}(\cdot), \text{Sum}(\cdot)$ and $\text{Max}(\cdot)$ are the most commonly used readout functions for GNNs. Xu et al. [195] show that GNNs equipped with $\text{Sum}(\cdot)$ readout have the same discriminative power as the Weisfeiler-Lehman Test. Zhang et al. [220] propose a sort pooling to generate the graph-level representation by sorting the final node representations. Ju et al. [104] 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 3.1(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,

$$\boldsymbol{P}_{ik} = \frac{e^{\langle \boldsymbol{Z}^L_{i\cdot}, \boldsymbol{E}_{k\cdot} \rangle}}{\sum_{k'}^{K} e^{\langle \boldsymbol{Z}^L_{i\cdot}, \boldsymbol{E}_{k'\cdot} \rangle}}, \tag{3.2}$$

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 3.1(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 [75] 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

$$\boldsymbol{u}_k = \boldsymbol{C}_{k\cdot} - \sum_{j=1}^{k-1} \frac{\langle \boldsymbol{u}_j, \boldsymbol{C}_{k\cdot} \rangle}{\langle \boldsymbol{u}_j, \boldsymbol{u}_j \rangle} \boldsymbol{u}_j, \quad \boldsymbol{E}_{k\cdot} = \frac{\boldsymbol{u}_k}{\|\boldsymbol{u}_k\|}. \tag{3.3}$$

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

Theoretical Justifications

In OCRead, proper cluster centers can generate higher-quality soft assignments and enlarge the difference between $\boldsymbol{P}$ from different classes. [161, 138] 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. (3.2):

Theorem 3.3.1. *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 feature vectors. Let V_r be the volume of B_r. The variance of Softmax projection averaged over B_r

$$\frac{1}{V_r}\int_{B_r}\sum_{k}^{K}\Big(\frac{e^{\langle \mathcal{Z},\boldsymbol{E}_{k\cdot}\rangle}}{\sum_{k'}^{K}e^{\langle \mathcal{Z},\boldsymbol{E}_{k'\cdot}\rangle}}-\frac{1}{K}\Big)^2 d\mathcal{Z}, \tag{3.4}$$

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 B.3.

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*:

$$\boldsymbol{P}_T(\boldsymbol{Z}_{i\cdot},\boldsymbol{E}_{k\cdot})=\boldsymbol{P}(\boldsymbol{Z}_{i\cdot},\boldsymbol{E}_{k\cdot})+\epsilon_i,\quad \epsilon_i\sim N(0,\sigma^2),\quad E(\epsilon_i)=0,\quad D(\epsilon_i)=\sigma^2, \tag{3.5}$$

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* [139], 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 3.3.2. *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 B.3.

3.3.4 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 [7]. 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 [219, 218]. 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 [142], whereas for gene expression networks, genes can be grouped based on the latent pathway information [110]. 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.

3.4 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?

3.4.1 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 [19]. 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 [42]. 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 B.1 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) [22]. 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 [74].

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 B.9 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 [126] and BrainGB [46].

3.4.2 RQ1: Performance Analysis

We compare BrainNetTF with baselines of three types. The details about how to tune hyperparameters of various baselines can be found in Appendix B.8. Besides, Appendix B.7 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 [120] and Graphormer [207]. 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.2 show that adding eigen node features to connection profiles cannot improve the model's performance. Besides, the benchmark paper [46] reveals that injecting edge weights into the attention mechanism can significantly reduce the prediction power. Furthermore, Appendix B.6 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 [126] designs ROI-aware GNNs for brain network analysis. BrainGB [46] is a systematic study of how to design effective GNNs for brain network analysis. We adopt their best design as the BrainGB baseline. BrainnetCNN [111] 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 3.1, BrainNetTF consistently outperforms BrainGNN, BrainGB and BrainnetCNN. **(c) BrainNetTF vs. neural network models on learnable brain networks.** Unlike classical GNNs, FBNETGEN [106], DGM [112] and BrainNetGNN [136] 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.

3.4.3 RQ2: Ablation Studies on the OCRead Module

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 vari-

Table 3.1: 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

ant. The results shown in Table 3.2 demonstrate that our OCRead is the most effective readout function for brain networks and improves the prediction power across various Transformer architectures.

Table 3.2: Performance comparison AUROC (%) with different readout functions.

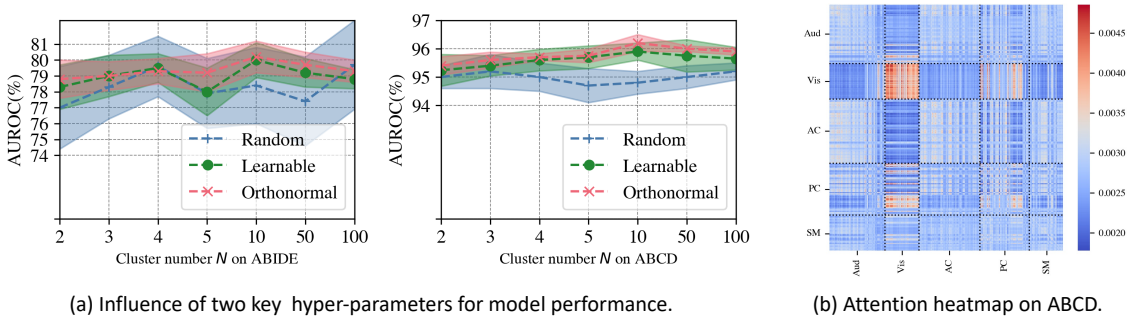
Readout	Dataset: ABIDE			Dataset: ABCD		
	SAN	Graphormer	VanillaTF	SAN	Graphormer	VanillaTF
MEAN	63.7±2.4	50.1±1.1	73.4±1.4	88.5±0.9	87.6±1.3	91.3±0.7
MAX	61.9±2.5	54.5±3.6	75.6±1.4	87.4±1.1	81.6±0.8	94.4±0.6
SUM	62.0±2.3	54.1±1.3	70.3±1.6	84.2±0.8	71.5±0.9	91.6±0.6
SortPooling	68.7±2.3	51.3±2.2	72.4±1.3	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**

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 [75] 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. (3.3).

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 3.3(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 3.3: The hyper-parameter influence and the heatmap from self-attention.

3.4.4 RQ3: In-depth Analysis of Attention Scores and Cluster Assignments

Figure 3.3(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 3.4 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 B.10, 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.

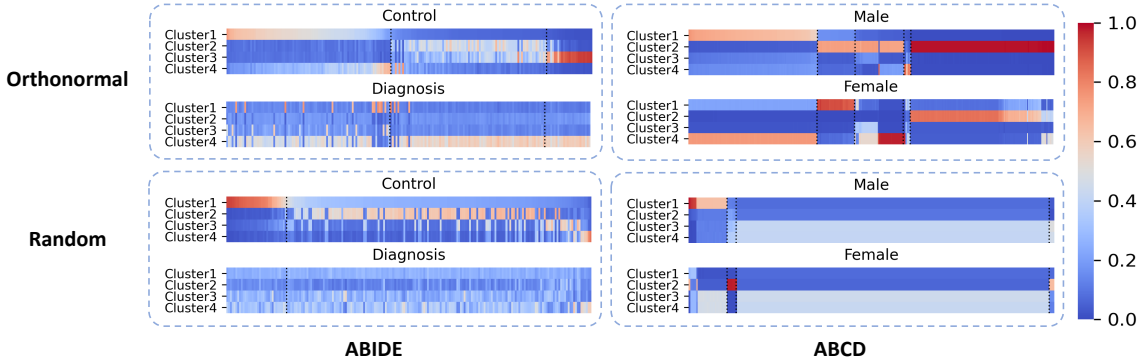


Figure 3.4: Visualization of cluster (module-level) embeddings learned with Orthonormal vs. Random cluster center initializations on two datasets. Each group in the dotted box contains two heatmaps (one for each prediction class) with the same node ordering on the x-axis.

3.5 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 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 present 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.

Chapter 4

Model Architecture: DynAmic bRain Transformer (DART) with Multi-level Attention for Functional Brain Network Analysis

4.1 Introduction

Network-centric analysis on brain imaging has gained substantial attention in neuroimaging studies recently, contributing profoundly to our understanding of brain organization in healthy individuals and those with brain disorders [18]. Neuroscience research has consistently demonstrated that insights into neural circuits are pivotal for distinguishing brain function across diverse populations, with disruptions in these circuits often instigating and delineating brain disorders [99]. Functional magnetic resonance imaging (fMRI) has emerged as a widely employed imaging modality for exploring brain function and organization [168]. Predicting clinical outcomes or categorizing individuals based on brain networks extracted from fMRI images with

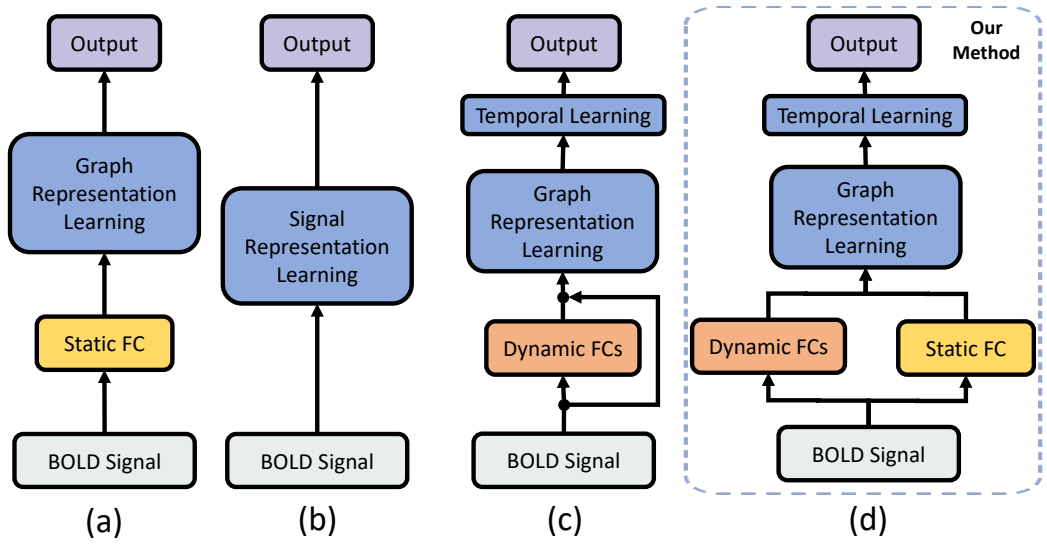


Figure 4.1: Four distinct schemas when employing Deep Learning for brain network analysis. From fMRI imaging, three types of input data can be acquired: (1) raw time-series data (BOLD signals), (2) static functional connectivity (FC), and (3) dynamic FCs, which capture temporal changes. Both kinds of FC are derived from the BOLD signal. Our method is the first attempt to combine Static FCs and dynamic FCs.

deep neural networks is a topic of significant interest in the neuroimaging community [111, 107, 44, 226].

Figure 4.1 succinctly summarizes various schemas used for analyzing brain networks with neural networks. The classic approach to network analyses primarily relies on using individual fMRI data to construct functional brain networks [170]. This established process involves selecting a brain atlas or regions of interest (ROI), extracting fMRI blood-oxygen-level-dependent (BOLD) signal series from each node or region, and computing pairwise connectivity measures. Once static brain networks are obtained, various neural network models can be applied for downstream analyses, as demonstrated in Figure 4.1 (a). There have been attempts to model BOLD signals directly using deep neural networks (DNNs), as seen in Figure 4.1 (b), but these have generally yielded unsatisfactory results due to the low signal-to-noise ratio of BOLD signals [106, 213]. However, recent works have attempted to use dynamic brain networks to replace static ones for downstream analyses [131, 114]. These dy-

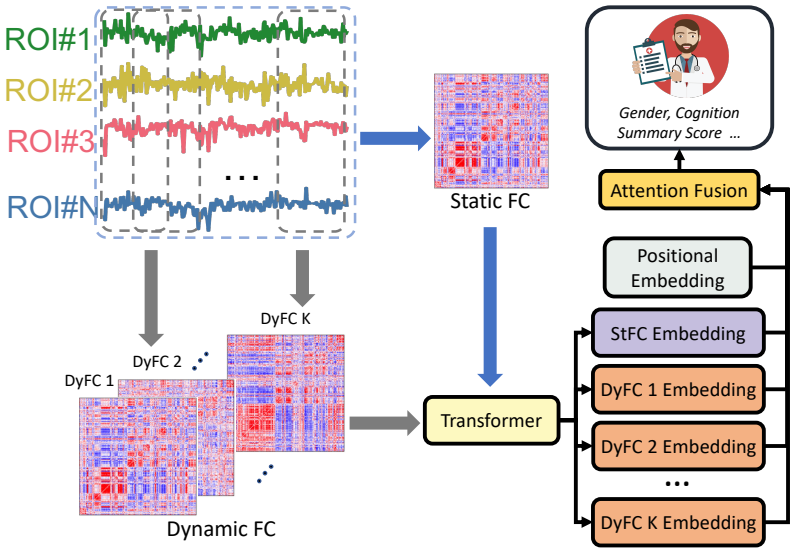


Figure 4.2: Diagram illustrating the comprehensive workflow of the proposed methodology, DART.

namic networks are created by segmenting BOLD signals into several overlapping or non-overlapping windows, each contributing to a unique connectivity matrix. This strategy shown in Figure 4.1 (c) allows for exploring temporal variations and state transitions in functional connectivity over time, providing crucial insights into brain function. However, there is significant space for improvement due to the high dimensionality and complexity of dynamic brain networks. In response to these challenges, this paper proposes a novel methodology, DynAmic bRain Transformer (DART), depicted in Figure 4.1 (d). DART exploits static brain networks as a foundation measurement to integrate dynamic brain networks, thereby improving performance against benchmark methods. Additionally, we incorporate specific attention mechanisms to enhance model explainability, aiming to capitalize on dynamic brain networks' switching in neuroimaging studies.

Table 4.1: Performance comparison with baselines. The ↑ indicates a higher metric value is better, while ↓ is opposite.

Type	Method	PNC		ABCD
		AUROC↑	Accuracy↑	MSE↓
Dynamic	STAGIN	63.5±4.0	54.2±1.4	102.4±6.1
	ST-GCN	64.7±3.5	57.3±3.2	89.2±11.2
Static	BrainGNN	62.4±3.5	59.4±2.3	80.8±4.7
	BrainGB	69.7±3.3	63.6±1.9	78.1±4.3
	BrainNetCNN	74.9±2.4	67.8±2.7	77.1±4.5
	BNT	78.2±1.9	70.6±2.1	60.2±1.5
Dynamic & Static	DART	**80.7±3.1**	**72.5±2.3**	**58.3±3.5**

4.2 Method

In this section, we elaborate on the design of DART and its four main components as shown in Figure 4.2. Specifically, the input $\boldsymbol{X} \in \mathbb{R}^{v \times T}$ denotes the BOLD time-series for regions of interest (ROIs) represents a sample (individual), v is the number of ROIs, and T is the length of time-series. We set L as the window size, S as the stride size. Given a sample $\boldsymbol{X}$, we can obtain k dynamic brain networks, where $k = \lfloor \frac{T-L+S}{S} \rfloor$. For the classification task, the target output is the prediction label $\boldsymbol{Y} \in \mathbb{R}^{|\mathcal{C}|}$, where $\mathcal{C}$ is the class set of Y and $|\mathcal{C}|$ is the number of classes. For the regression task,the target output is the prediction label $\boldsymbol{Y} \in \mathbb{R}$.

Static FC Generation. We begin by generating a static functional connectivity (FC) matrix, which provides a summary of the overall functional connections in the brain during the entire scan period. This static FC, $\boldsymbol{A} \in \mathbb{R}^{v \times v}$, represents the connectivity matrix between all pairs of ROIs for each individual. Specifically, we use the Pearson Correlation as a measure of statistical dependence between the time series of different ROIs. Each element of the static FC, $\boldsymbol{A}_{ij}$, is computed as $\text{Corr}(\boldsymbol{X}_i, \boldsymbol{X}_j)$, which denotes the correlation between the time-series of ROI i and ROI j. This matrix captures the overall brain functional organization and serves as an anchor for the subsequent steps.

Dynamic FC Generation. To generate dynamic brain networks, we partition the BOLD signal into a series of overlapping or non-overlapping windows of length T, with a stride of size S. We calculate a dynamic functional connectivity matrix for the time window t, $\boldsymbol{D}^t \in \mathbb{R}^{v \times v}$. Finally, we can obtain k dynamic functional connectivity matrix, where k is the total number of dynamic networks given by $k = \lfloor \frac{T-L+S}{S} \rfloor$. Each of these matrices represents a snapshot of brain connectivity at a specific time point. Similar to the static FC generation, we use Pearson Correlation for this computation. Each element in the dynamic connectivity matrix, $\boldsymbol{D}^t_{ij}$, is then calculated as $\text{Corr}(\boldsymbol{X}^t_i, \boldsymbol{X}^t_j)$, where $\boldsymbol{X}^t_i$ and $\boldsymbol{X}^t_j$ are the BOLD signals for ROIs i and j at the time window t.

Edge-level Attention and Transformer Projection. After generating both static and dynamic FCs, we utilize the graph transformer proposed by [107] for processing these matrices. This transformer comprises a Multi-Head Self-Attention Module, which is adept at capturing complex dependencies between different nodes in the network, thus enabling a rich representation of the FCs. A learnable clustering readout function is applied to compress the matrix into a graph-level embedding. In the case of the static brain network, the hidden representation, $\boldsymbol{h}_A = \boldsymbol{f}_{\text{TF}}(A)$, is obtained. In contrast, for each dynamic brain network, the hidden representation is equipped with an attention layer $\boldsymbol{\alpha}$ and added by a positional embedding as delineated in [182], resulting in $\boldsymbol{h}^t_D = \boldsymbol{f}_{\text{TF}}(\boldsymbol{\alpha} \circ \boldsymbol{D}^t) + \boldsymbol{P}^t$. Here, $\circ$ denotes the Hadamard product, $\boldsymbol{\alpha} \in \mathbb{R}^{v \times v}$ represents a learnable attention matrix (initialized at 1) shared across all dynamic networks, and $\boldsymbol{P}^t$ refers to the positional embedding at time window t. The attention mechanism based on $\boldsymbol{\alpha}$ allows the model to focus on the most informative connections in the brain networks.

Temporal-level Attention and Fusion. Given the hidden embedding of static FC $\boldsymbol{h}_A$ and the sequence of dynamic FCs' hidden embedding $\boldsymbol{h}^t_D$, we utilize an attention mechanism to fuse these networks. The attention scores are computed based

on the similarity between $\boldsymbol{h}_A$ and each $\boldsymbol{h}_D^t$. Specifically, the attention score $\boldsymbol{\beta}^t$ for each dynamic FC $\boldsymbol{h}_D^t$ is calculated as $\boldsymbol{\beta}^t = \text{softmax}(\text{sim}(\boldsymbol{h}_A, \boldsymbol{h}_D^t))$, where sim(.) is a similarity function, such as dot product. Then, the fused FC $\boldsymbol{F}$ is generated as a weighted sum of dynamic FCs, i.e., $\boldsymbol{F} = \sum_{t=1}^{k} \boldsymbol{\beta}^t \cdot \boldsymbol{h}_D^t$. The final FC $\boldsymbol{F}$ is then fed into the multi-layer perception module for final prediction. The fusion mechanism thus enables the model to direct its focus towards dynamic FCs bearing higher similarity to the static FC, effectively leveraging the static FC's stable functional information to guide the dynamic FC fusion.

4.3 Experiments

4.3.1 Experimental Settings

Dataset. This study utilizes two public neuroimaging datasets. The first is the Adolescent Brain Cognitive Development Study (ABCD), one of the largest publicly available fMRI datasets with stringent access control [22]. We employ fully anonymized brain networks based on the HCP 360 ROI atlas [74] and define a task, the Cognition Summary Score Prediction, a regression problem focused on five cognitive sub-domains. Considering the variability in sequence lengths within the ABCD dataset, we included only those samples with a sequence length exceeding 1024, truncating them to form a unified length dataset, yielding 4613 samples for regression analysis. The second dataset is the Philadelphia Neuroimaging Cohort (PNC), whose individuals aged 8–21 years provided by the Children's Hospital of Philadelphia [159]. After quality control, the dataset includes 503 subjects, each providing 120 timesteps of data from 264 nodes [152].

Metric. To evaluate performance in binary classification tasks, we employ two widely accepted metrics: Area Under the Receiver Operating Characteristic (AUROC) and accuracy. We set the classification threshold at 0.5 to determine the final class labels.

In the case of regression tasks, we utilize the Mean Square Error (MSE) as a comprehensive measure of model performance. Please note, all the results presented in this study are the mean values derived from five independent runs, each initiated with a different random seed, to ensure the robustness and reproducibility of our findings.

Implementation. We configure the window size L and stride size as 24 to ensure each window encapsulates a one-minute BOLD signal. The architecture of our Transformer is built according to the design described in [107], setting the number of transformer layers to 2, matching the hidden dimension for each transformer layer with the number of nodes v, and employing 4 heads. We divide our datasets such that 70% is utilized for training, 10% for validation, and the remainder for testing. We leverage the Adam optimizer throughout the training process with a learning rate and weight decay set at 10^{-4}. Our batch size is 16, and all models undergo 200 training epochs. The epoch displaying optimal performance on the validation set is chosen for the final report.

4.3.2 Performance and Analysis

Our model's performance is benchmarked against several state-of-the-art methodologies in brain network analysis, and the result can be found in Table 4.1. We consider methods that leverage both static and dynamic brain networks for comparison. The baseline models include STAGIN [114], which constructs dynamic brain networks and fuses them using an attention mechanism without considering static brain networks; ST-GCN [71], an improved version of GCN that takes into account not only the current graph but also the adjacency of prior and future graphs; BrainGNN [126] and BrainGB [46], two Graph Neural Networks designed explicitly for static brain networks; BrainNetCNN [111], a convolutional neural network model designed for static brain networks; and finally BNT [107], a graph transformer model also designed for static brain networks, which is the same transformer that we employ in our model to project brain networks into an embedding.

The comparative analysis with these methods demonstrates several vital insights. Models that rely exclusively on dynamic brain networks perform the poorest, underlining the critical role of global information provided by static brain networks in predictions. Static brain networks were found to encompass the most significant predictive signals, which illustrates our strategy of using the hidden representations of these static networks as anchor points (or query embeddings) to fuse dynamic brain representations. Furthermore, integrating dynamic brain networks is observed to enhance model performance because it exploits the fine-grained variations in brain states. As a result, our proposed methodology, denoted as DART, consistently outperforms all baseline methods, exhibiting the highest performance across various metrics on two datasets with both regression and classification tasks.

4.3.3 Attention Visualization and Analysis

Due to the special attention design in DART, the proposed method enables two-level attention-driven interpretability - edge-level attention ($\boldsymbol{\alpha}$), representing the significance of each edge, and temporal-level attention ($\boldsymbol{\beta}$), indicating the importance of each dynamic brain network. As evidenced in Figure 4.3, during the first 100 epochs, attention weights progressively evolve, stabilizing over the subsequent 100 epochs. The edge-level attention, initially uniform, progressively concentrates on the Visual and Default Mode sub-networks, aligning with the scores evaluated from heavy visual tasks like the Picture Vocabulary and Picture Sequence Memory Tests. Meanwhile, the temporal-level attention highlights dynamic brain networks recorded during the middle collection. Given the fact that the ABCD dataset is collected from children, it is plausible that resting-state brain activity is most prominent for children during the middle of the data collection process, corroborating our temporal attention insights.

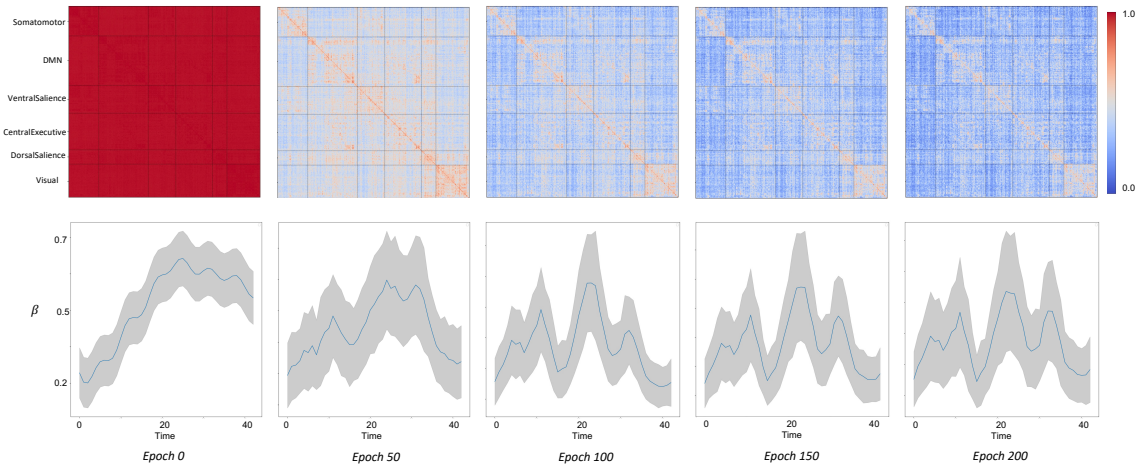


Figure 4.3: Evolution of two-level attention during the training on the ABCD dataset. The first row displays the progression of edge-level attention ($\boldsymbol{\alpha}$) across epochs, while the second row shows the changes in temporal-level attention ($\boldsymbol{\beta}$) across epochs.

4.4 Conclusion

Our proposed method, DART, addresses the challenges presented by dynamic brain networks' complexity and high dimensionality, thereby advancing brain network analysis in neuroimaging. By leveraging static brain networks as a foundational measurement, we successfully integrate dynamic brain networks and improve performance compared to standard methods. The specific attention mechanisms incorporated within DART further enhance model explainability.

Chapter 5

Data Augmentation: Riemannian Mixup for Improved Generalization

5.1 Introduction

As a ubiquitous type of data in biomedical studies, biological networks are used to depict a complex system with a set of interactions between various biological entities. For example, in a brain network, the correlations extracted from functional Magnetic Resonance Imaging (fMRI) are modeled as interactions among human-divided brain regions [170, 167, 190, 129, 29, 106, 93, 94]. Meanwhile, in a co-expression gene-protein network, interactions are built to discover disease genes and potential modules for clinical intervention [146]. There are diverse ways to define the connections among entities in biological networks, such as interactions [210, 127], reactions [40], and relations [105, 47, 196, 46, 44]. One of the most widespread practices is calculating the covariance and correlation among entities to summarize and quantify interactions [15, 167, 209, 58, 188, 187]. Therefore, developing powerful computational methods to predict disease outcomes based on profiling datasets from such correlation matrices has attracted great interest from biologists [126, 200, 4, 211, 125, 128, 204, 103].

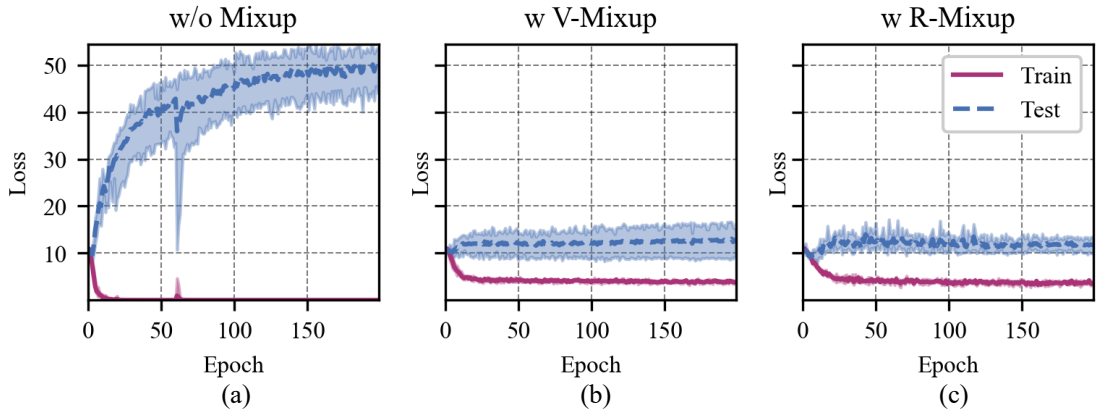


Figure 5.1: Train/Test performance of a Transformer on the biological network dataset PNC with 503 samples. Each sample is represented as a 120×120 adjacency matrix. V-Mixup is the vanilla Mixup and R-Mixup is our proposed method.

Deep learning methods have achieved state-of-the-art performance in various downstream applications [35, 111], especially when the training sample size is large enough. However, biological network datasets often suffer from limited samples due to the complicated and expensive collection and annotation processes of scientific data [22, 222, 198]. Another key property of biological networks is that the dimension of such networks is typically very high, i.e., $O(n^2)$ correlation edges among n entities. Therefore, directly applying Deep Neural Networks (DNNs) to such biological network datasets can easily cause severe overfitting [5, 197, 212, 50, 203].

Mixup is a widely used data augmentation technique that can improve the model performance by linearly interpolating new samples from pairs of existing instances [217]. In the scenario of biological network analysis, since the node identities and their corresponding order are usually fixed across network samples within the same dataset [107], the Mixup technique can be easily applied via linear interpolation. Empirically, Figure 5.1 (a) and (b) compare the processes of training a transformer model [107] without Mixup and with the vanilla Mixup (V-Mixup) [217] technique on the brain network dataset from the PNC studies [159] to perform binary classification. In Figure 5.1 (a), the training loss without the Mixup technique diminishes quickly while

the test loss continues to increase, which apparently indicates a severe overfitting problem. In contrast, in Figure 5.1 (b) with V-Mixup, the training process becomes more stable, and the model achieves higher performance with a lower test loss, even though the training loss is relatively high.

Although the vanilla Mixup can mitigate the overfitting issue for biological networks, there are two critical limitations in existing Mixup methods. The first noticeable issue is that the linear Mixup of correlation matrices in the Euclidean space would cause a *swelling effect*, where the determinant of the interpolated matrix is larger than any of the original ones. The inflated determinant, which equals the product of eigenvalues, also indicates an increase in eigenvalues. This can be interpreted as exaggerated variances of the data points in the principal eigen-directions. As a result, an unphysical augmentation from original data is generated, which may change the characteristics, e.g., the correlations of different brain functional areas, of the original dataset and violate the intuition of linear interpolations that the determinant from a mixed sample should be intuitively *between* the original pair of samples [24, 62, 150, 57]. On the other hand, the vanilla Mixup cannot properly handle regression tasks due to *arbitrarily incorrect label* [205], which means that linearly interpolating a pair of examples and their corresponding labels cannot ensure that the synthetic sample is paired with the correct label. Although several existing works like RegMix [98] and C-Mixup [205] have attempted to avoid this issue by restricting the mixing process only to samples with a similar label, their practice leads to less various sample generation and weakens the ability of Mixup towards improving the robustness and generalization of deep neural network models.

Recently, investigating covariance and correlation matrices in the view of symmetric positive definite matrices (SPD) with Riemannian manifold has demonstrated impressive advantages in biological domains [9, 208, 147], which helps to improve the model performance and capture informative sample features. Inspired by these stud-

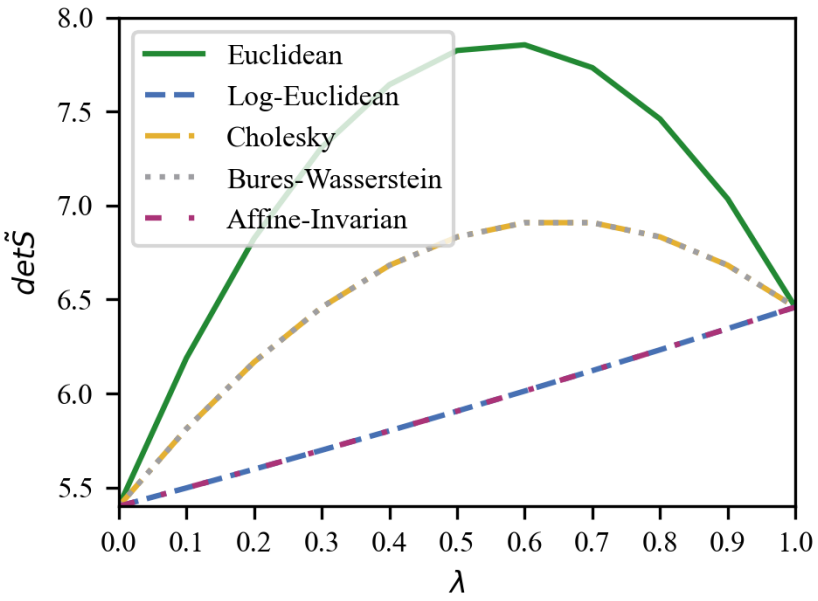


Figure 5.2: The *swelling effect* of Mixing up with different metrics. $\tilde{S}$ is the augmented sample mixed by samples S_i and S_j, where $\det S_i = 5.40$ and $\det S_j = 6.46$. Ideally, the determinant of the mixed sample $\tilde{S}$ should be between $\det S_i$ and $\det S_j$. The results indicate that mixing samples with Euclidean (widely used in existing Mixup methods), Cholesky, and Bures-Wasserstein metrics leads to unphysical inflations.

ies, we pinpoint a promising direction to mitigate these two identified issues when adapting the Mixup technique for biological networks from the perspective of SPD analysis. However, existing works that leverage the Riemannian manifold for SPD analysis of biological networks often directly treat covariance and correlation matrices as SPD matrices without rigorous verification. We clarify that covariance and correlation matrices are not equal to SPD matrices: a necessary condition for the covariance and correlation matrices generated from a sample $X \in \mathbb{R}^{n \times t}$ to be positive definite is that *the sequence length t is no less than the sample variable number n*. We provide theoretical proof for this condition in Appendix C.1. The collection of positive definite matrices mathematically forms a unique geometric structure called the *Riemannian manifold*, which generalizes curves and surfaces to higher dimensional objects [124, 72, 158]. From a mathematical perspective, augmenting samples along geodesics on the manifold of SPDs with the log-Euclidean metric effectively (a) preserves the intrinsic geometric structure of the original data and eases the *arbitrarily incorrect label* and (b) eliminates the *swelling effect* as is shown in Figure 5.2. The advantages are further proved theoretically in Section 5.3.

Based on this insight, we propose R-Mixup, a Mixup-based data augmentation approach for SPD matrices of biological networks, which augments samples based on Riemannian geodesics (i.e., Eq.(5.2)) instead of straight lines (i.e., Eq.(5.1)). We theoretically analyze the advantages of R-Mixup by incorporating tools from differential geometry, probability and information theory. Besides, a simple and efficient preprocess optimization is proposed to reduce the actual training time of R-Mixup considering the costly eigenvalue decomposition operation. Sufficient experiments on five datasets spanning both regression and classification tasks demonstrate the superior performance and generalization ability of R-Mixup. For regression tasks, R-Mixup can achieve the best performance based on the same random sampling strategy as vanilla Mixup, demonstrating its ability to overcome the *arbitrarily in-*

correct label issue by adequately leveraging the intrinsic geometric structure of SPD. This advantage is also proved by a case study in Appendix C.7. Furthermore, we observe that the performance gain of R-Mixup over existing methods is especially prominent when the annotated samples are extremely scarce, verifying its practical advantage under the low-resource settings.

We summarize the contributions of this work as three folds:

- We propose R-Mixup, a data augmentation method for SPD matrices in biological networks, which leverages the intrinsic geometric structure of the dataset and resolves the *swelling effect* and *arbitrarily incorrect label* issues. Different Riemannian metrics on manifold are compared, and the effectiveness of R-Mixup is theoretically proved from the perspective of statistics. We also proposed a pre-computing optimization step to reduce the burden from eigenvalue decomposition.
- Thorough empirical studies are conducted on five real-world biological network datasets, demonstrating the superior performance of R-Mixup on both regression and classification tasks. Experiments on low-resource settings further stress its practical benefits for biological applications often with limited annotation.
- We emphasize a commonly ignored necessary condition for viewing covariance and correlation matrices as SPD matrices. We believe the clarification of this prerequirement for applying SPD analysis can enhance the rigor of future studies.

5.2 Related Work

5.2.1 Mixup for Data Augmentation

Mixup is a simple but effective principle to construct new training samples for image data by linear interpolating input pairs and forcing the DNNs to behave linearly in-between training examples [217]. Many follow-up works extend Mixup from different perspectives. For example, [186, 185] interpolate training data in the feature space,

[79, 34] learn the mixing ratio for Mixup to alleviate the under-confidence issue for predictions. Besides, [48, 97, 223, 205] strategically select the sample pairs for Mixup to prevent low-quality mixing examples and produce more reasonable augmented data. To further improve the quality of the augmented data, [214, 116, 86] create mixed examples by only interpolating a specific region (often most salient ones) of examples. Mixup has also been extended to other data modalities such as text [26, 221] and audio [140]. There are several attempts to study Mixup on non-Euclidean data, graphs, like NodeMixup [191], GraphMixup [193] and G-Mixup [82]. However, less attention has been paid to adapting Mixup for graphs from a manifold perspective, which is the focus of this study.

5.2.2 Geometric Deep Learning

Geometric deep learning aims to adapt commonly used deep network architectures from euclidean data to non-euclidean data, such as graphs and manifolds, with a broad spectrum of applications from the domains of radar data processing [17], graph analysis [143, 6], image and video processing [95, 23, 143, 85], and Brain-Computer Interfaces [174, 147]. For example, SPDNet [95] builds a Riemannian neural network architecture with special convolution-like layers, rectified linear units (ReLU)-like layers, and modified backpropagation operations for the non-linear learning of SPD matrices. ManifoldNet [23] defines the analog of convolution operations for manifold-valued data. MoNet [143] generalizes CNN architectures to the non-Euclidean domain with pseudo-coordinates and weight functions. [17] designs a Riemannian batch normalization for SPD matrices by leveraging geometric operations on the Riemannian manifold. MAtt [147] proposes the manifold attention mechanism to represent spatiotemporal representations of EEG data. Though widely recognized as being effective for images, tabular and graph data, to the best of our knowledge, data augmentation methods in geometric deep learning have rarely been explored.

5.3 R-Mixup

In this section, we first provide some preliminary facts, including a necessary condition for treating covariance and correlation matrices as SPD matrices. Next, we elaborate on the detailed process of applying R-Mixup for data augmentation, compare possible mathematical metrics designs, and finally provide the theoretical analysis of the advantages of using R-Mixup.

5.3.1 Notations and Preliminary Results

Given n variables of biological entities, we extract a t length sequence for each variable and compose the input sequences $X \in \mathbb{R}^{n\times t}$. The correlation matrix or biological network $S = \mathrm{Cor}(X) \in \mathbb{R}^{n\times n}$ is obtained by taking the pairwise correlation among each pair of the biological variables. The value y is the network-level prediction label for the prediction task.

Definition 5.3.1. A symmetric $n \times n$ matrix S is *positive semi-definite* if for any vector $u \in \mathbb{R}^n$, $u^T S u \geq 0$. Equivalently, this means that the eigenvalues of S are all nonnegative. If the inequality holds strictly, S is said to be *positive definite*, or *symmetric positive definite*, or SPD for short.

Let $\mathrm{Sym}(n)$ be the collection of all positive semi-definite matrices, and $\mathrm{Sym}^+(n)$ denotes the collection of all SPDs. The collection $\mathrm{Sym}(n)$ can be seen as an $\frac{1}{2}n(n-1)$-dimensional Euclidean space, but $\mathrm{Sym}^+(n) \subset \mathbb{R}^{n\times n}$ admits a more general structure call *manifold* in differential geometry which resembles the Euclidean space in its local regions. To set up the modeling on the manifold $\mathrm{Sym}^+(n)$, the covariance matrix $\mathrm{Cor}(X)$ for the input X should be positive definite. However, it is worth mentioning that previous studies that use the Riemannian manifold for analyzing biological networks often treat covariance and correlation matrices as SPD without proper validation. Towards this common negligence, we bring out the following basic

fact:

Proposition 5.3.2. *Covariance and correlation matrices are positive semi-definite. A necessary condition for them to be positive definite is that the sample length is no less than the variable number, i.e., $t \geq n$.*

This proposition indicates that covariance and correlation matrices only have the opportunity to be positive definite when $t \geq n$. The detailed proof can be found in Appendix C.1. This is the case for the datasets involved in this study, where most of the correlation matrices are SPD. The few exceptions would have very few zero eigenvalues, which we manually set as 10^{-6} to eliminate their influence. More discussions on adjusting correlation matrices to be SPDs can be found in [39, 92].

5.3.2 R-Mixup Deduction

In this section, we explain on the detailed process of R-Mixup for SPD matrice augmentation. Let S_i, S_j represent two different correlation matrices constructed based on X_i, X_j. In the vanilla Mixup [217], the augmented samples $(\tilde{S}, \tilde{y})$ are created through the straight line connecting S_i, S_j and y_i, y_j,

$$\begin{aligned} \tilde{S} &= (1-\lambda)S_i + \lambda S_j, \\ \tilde{y} &= (1-\lambda)y_i + \lambda y_j, \end{aligned} \tag{5.1}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$, Beta is the Beta distribution, given $\alpha \in (0, \infty)$.

To facilitate the illustration of R-Mixup in geometry notions, we briefly introduce the main concepts here while more detailed explanations can be found in [124, 72]. To define R-Mixup, we replace Eq.(5.1) by a certain Riemannian geodesics. *Geodesics* are the generalization of *straight lines* in the Euclidean space, which is intuitively the shortest path between two given points on Riemannian manifolds. Riemannian manifolds (M, g) are manifolds M equipped with *Riemannian metrics* g which measure

distances between points in the manifold and induces geodesic equations [124, 72]. It is generally hard to solve geodesics equations in the simple analytical form as straight lines, however, for $\mathrm{Sym}^+(n)$, there are lots of well-defined choices of Riemannian metrics with known geodesics [150, 100, 57, 72], and we employ the *log-Euclidean metric* with the following geodesic:

$$\tilde{S} = \exp\left((1-\lambda)\log S_i + \lambda \log S_j\right), \tag{5.2}$$

where $\exp, \log$ are matrix exponential and logarithm. Figure 5.3 sketches the geodesic as the purple dotted curve and a rigorous deduction of Eq.(5.2) can be found in [72]. Implementation of the matrix exponential for positive definite matrix S is straightforward: by basic linear algebra,

$$S = O\mathrm{diag}(\mu_1, ..., \mu_n)O^T, \tag{5.3}$$

where O is an orthogonal matrix with μ_i being eigenvalues of S. Then by definition,

$$\begin{aligned} \exp S &= O\mathrm{diag}(\exp\mu_1, ..., \exp\mu_n)O^T, \\ \log S &= O\mathrm{diag}(\log\mu_1, ..., \log\mu_n)O^T. \end{aligned} \tag{5.4}$$

5.3.3 Comparison with Other Metrics

There are various choices of Riemannian metrics and hence different geodesics on $\mathrm{Sym}^+(n)$ [150, 57, 100, 14, 72], such as the Cholesky metric defined by Cholesky decompositions L_i of positive definite matrices S_i, the well-known Affine-invariant metrics on $\mathrm{Sym}^+(n)$ [176], and the Bures-Wasserstein studied in statistics and information theory [14, 13]. We compare the most popular ones with the proposed log-Euclidean for mixing up biological networks on prediction tasks. The compar-

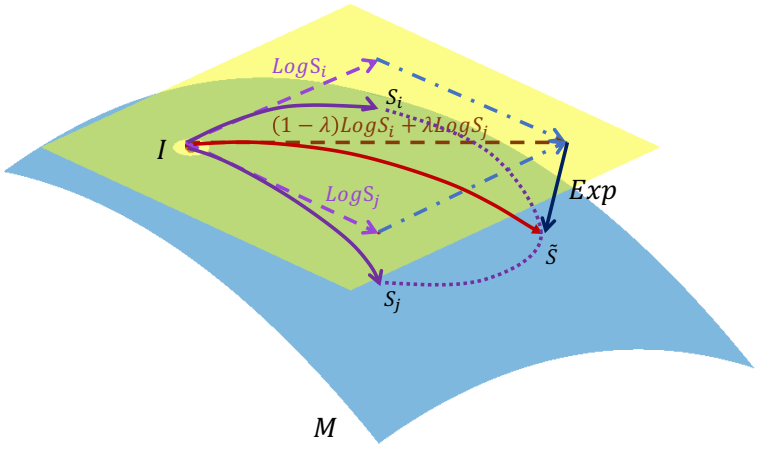


Figure 5.3: The process of R-MIXUP generating sample $\tilde{S}$, where the blue surface M represents the *Riemannian manifold* and the yellow plane is the tangent plane of M at the origin I. S_i, S_j are the original samples in M, and $\log S_i, \log S_j$ are tangent vectors. R-MIXUP creates the augmented sample $\tilde{S}$ by combining the initial tangent vectors of both trajectories connecting I with S_i, S_j, i.e., $(1-\lambda)\log S_i + \lambda \log S_j$, and push it back to the *Riemannian manifold* M via exponential map.

isons are summarized in Table 5.1. To be specific, different geodesics are analyzed from two perspectives: (a) whether it causes the swelling effect, (b) whether it is numerically stable on our dataset.

Swelling Effect. The detailed definition and rigorous proof of the swelling effect can be found in Section 5.3.4 and Appendix C.2. As exemplified by the motivation in Figure 5.2, Euclidean, Cholesky, and Bures-Wasserstein metrics evidently suffer from the swelling effect.

Numerical Stability. Augmenting matrices from the geodesic with the Affine-invariant metric requires the computation of $S_i^{-1/2}$ and hence the calculation of the inverse square root of its eigenvalues as we define matrix exponential and logarithm in Eq.(5.4). For SPDs with small eigenvalues μ, such computations may not be numerically stable since $\mu^{-1/2} \to \infty$. Furthermore, with awareness to the following

limit relation:

$$\lim_{\mu \to 0} \frac{\log \mu}{\mu^{-1/2}} = 0, \tag{5.5}$$

which indicates that $\log \mu \ll \mu^{-1/2}$ for small μ, we know that computing matrix logarithm when using log-Euclidean metric should be more stable. Similarly, for Bures-Wasserstein geodesics, to compute $(S_i S_j)^{1/2}$, we notice the following fact:

$$S_i S_j = S_i^{1/2}\Big(S_i^{-1/2}(S_i S_j)S_i^{1/2}\Big)S_i^{-1/2} = S_i^{1/2}\Big(S_i^{1/2} S_j S_i^{1/2}\Big)S_i^{-1/2}. \tag{5.6}$$

Thus,

$$(S_i S_j)^{1/2} = S_i^{1/2}\Big(S_i^{1/2} S_j S_i^{1/2}\Big)^{1/2} S_i^{-1/2}, \tag{5.7}$$

where the undesirable $\mu^{-1/2}$ appears again in the calculation.

Considering these two points, we stick with log-Euclidean metric. Experimental results in Section 5.4.2 further showcase the effectiveness of this choice.

Table 5.1: Comparison of Different Metrics Choices

Metric	**Geodesics**	**Swelling Effect**	**Numerical Stability**
Euclidean	$(1-\lambda)S_i + \lambda S_j$	Yes	Stable
Cholesky	$((1-\lambda)L_i + \lambda L_j)((1-\lambda)L_i + \lambda L_j)^T$	Yes	Stable
Bures-Wasserstein	$(1-\lambda)^2 S_i + \lambda^2 S_j + \lambda(1-\lambda)((S_i S_j)^{1/2} + (S_j S_i)^{1/2})$	Yes	Unstable
Affine-invariant	$S_i^{1/2}(S_i^{-1/2} S_j S_i^{-1/2})^{\lambda} S_i^{1/2}$	No	Unstable
Log-Euclidean	$\exp((1-\lambda)\log S_i + \lambda \log S_j)$	**No**	**Stable**

5.3.4 R-Mixup Theoretical Justification

Using geodesics when conducting data augmentation demonstrates unique advantages over straight lines. The first advantage is that R-Mixup will not cause the *swelling effect* which exaggerates the determinant and certain eigenvectors of the samples as discussed in Section 5.1 and 5.3.3. Mathematically, suppose $\det S_i \leq \det S_j$, then the

determinant of $\tilde{S}$ defined by Eq.(5.2) satisfies:

$$\det S_i \leq \det \tilde{S} \leq \det S_j. \tag{5.8}$$

Detailed proof can be found in Appendix C.2.

The second advantage is that, by leveraging the manifold structure, we can fit better estimators compared with linear interpolation in the Euclidean space. To be precise, as illustrated after Proposition 5.3.2, our samples are distributed over $\mathrm{Sym}^+(n)$ rather than the whole ambient Euclidean space $\mathbb{R}^{n\times n}$, which is accepted as a *prior knowledge* in the sense of Bayesian modeling fitting. Then the purpose of implementing R-Mixup becomes clear: we augment the nontrivial geometric information for the learning architectures later used in our experiments as an analogy to transforming images to enhance the translation and rotational invariance before a training of image identification [32]. We theoretically justify this point from the perspective of both statistics on Riemannian manifolds [16, 115, 100, 101, 55] and information theory [21, 145, 154].

Specifically, we treat the data augmentation process as a regression conducted on the manifold $\mathrm{Sym}^+(n)$ which is explicitly constrained by its geometric structure and based on the distribution of the dataset as the prior knowledge. Given any $\tilde{S}$, let $\tilde{m}(\tilde{S})$ denote the *estimator/prediction function* of the regression whose analytical form depends on the concrete regression methods. We take geodesic regression and kernel regression [16, 66, 164] on $\mathrm{Sym}^+(n)$ to address the problem. Roughly speaking, geodesic regression generalizes multi-linear regression on Euclidean space to manifold with the Euclidean distance being replaced by Riemannian metric. Kernel regression embeds data into higher dimensional feature space with kernel functions K to grasp more non-linear relationship of the dataset. Since the exact distribution of augmented

data is unknown, we follow the common practice [83, 33, 164] and apply *Gauss kernel*

$$K_E(S_i, \tilde{S}) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{1}{2\sigma^2}\|S_i - \hat{S}\|^2\right), \tag{5.9}$$

which possess the *universal property* to approximate any continuous bounded function in principle. However, the Gauss kernel K_E is defined on the Euclidean space which *unreasonably* implies non-zero density of samples outside $\mathrm{Sym}^+(n)$ contradicting the prior knowledge. To remedy the problem, we introduce a method from the *heat kernel theory* in differential geometry [12, 72] to generalize K_E to

$$K_R(S_i, \hat{S}) = \frac{1}{(2\pi\sigma^2)^{\frac{n(n-1)}{4}}} \exp\left(-\frac{1}{2\sigma^2} d(S_i, \hat{S})^2\right), \tag{5.10}$$

with

$$d(S_i, S_j) = \|\log S_i - \log S_j\| \tag{5.11}$$

being the *Riemannian distances function* on $\mathrm{Sym}^+(n)$. Then we prove in details in the Appendix C.3.

Theorem 5.3.3. *For $Sym^+(n)$ with log-Euclidean metric, comparing* R-Mixup *with estimators $\tilde{m}$ obtained by regressions with respect to the manifold structure, the square loss for augmented data $\tilde{S}$ from Riemannian geodesics Eq.*(5.2) *is no more than those $\tilde{S}'$ from straight lines Eq.*(5.1)*:*

$$\sum(\tilde{m}(\tilde{S}) - \tilde{y})^2 \leq (\tilde{m}(\tilde{S}') - \tilde{y})^2. \tag{5.12}$$

A less empirical loss from regression on manifold is recognized as an evidence that R-Mixup captures some geometric features of $\mathrm{Sym}^+(n)$, thereby providing the learning algorithm an opportunity to learn this feature. Finally, the proposed R-

MIXUP is formally defined as

$$\begin{aligned} \tilde{S} &= \exp\left((1-\lambda)\log S_i + \lambda \log S_j\right), \\ \tilde{y} &= (1-\lambda)y_i + \lambda y_j, \end{aligned} \tag{5.13}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$, for $\alpha \in (0, \infty)$.

5.3.5 Time Complexity and Optimization

One potential concern of the proposed R-MIXUP lies in its time-consuming operations of the eigenvalue decomposition and matrix multiplication (with the time complexity of $\mathcal{O}(n^3)$), which dominate the overall running time of R-MIXUP. In practice, we find that most common modern deep learning frameworks such as PyTorch [149] have been optimized for accelerating matrix multiplication. Thus, the main extra time consumption of the R-MIXUP is the exp and log operations of the three eigenvalue decompositions. We propose a sample strategy to optimize the running time of R-MIXUP by precomputing the eigenvalue decomposition and saving the orthogonal matrix O and eigenvalues $\{\mu_1, ..., \mu_n\}$ of each sample. This precomputing process can reduce the three computations of eigenvalue decomposition to once for each sample. Formally,

$$\begin{aligned} \tilde{S} = \exp\Big((1-\lambda)O_i \text{diag}(\log \mu_1, ..., \log \mu_n)O_i^T \\ +\lambda O_j \text{diag}(\log \nu_1, ..., \log \nu_n)O_j^T\Big), \end{aligned} \tag{5.14}$$

where $O_i \text{diag}(\mu_1, ..., \mu_n)O_i^T$ and $S_j = O_j \text{diag}(\nu_1, ..., \nu_n)O_j^T$ are the eigenvalue decompositions of S_i and S_j, respectively. The efficiency of this optimization is further discussed in Section 5.4.4.

5.4 Experiments

We evaluate the performance of R-Mixup comprehensively on real-world biological network datasets with five tasks spanning classification and regression. The dataset statistics are summarized in Table 5.2. The empirical studies aim to answer the following three research questions:

- **RQ1**: How does R-Mixup perform compared with existing data augmentation strategies on biological networks with various sample sizes on different downstream tasks?
- **RQ2**: How does the sequence length of each sample affect the characteristics of correlation matrices and consequently the choice of augmentation strategies?
- **RQ3**: Is R-Mixup efficient in the training process and robust to hyperparameter changes?

Table 5.2: Dataset Summary.

Dataset	Sample Size	Variance Number (n)	Sequence Length (t)	Task	Class Number
ABCD-BioGender	7901	360	Variable Length	Classification	2
ABCD-Cog	7749	360	Variable Length	Regression	-
PNC	503	120	120	Classification	2
ABIDE	1009	100	100	Classification	2
TCGA-Cancer	240	50	50	Classification	24

5.4.1 Experimental Setup

Datasets and Tasks

Adolescent Brain Cognitive Development Study (ABCD). The dataset used in this study is one of the largest publicly available fMRI datasets, with access restricted by a strict data requesting process [22]. From this dataset, we define two tasks: *BioGender Prediction* and *Cognition Summary Score Prediction*. The data used in the experiments are fully anonymized brain networks based on the HCP 360 ROI atlas [74] with only biological sex labels or cognition summary scores. BioGender

Prediction is a binary classification problem, which includes 7901 subjects after the quality control process, with 3961 (50.1%) females among them. Cognition Summary Score Prediction is a regression task whose label is Cognition Total Composite Score containing seven computer-based instruments assessing five cognitive sub-domains: Language, Executive Function, Episodic Memory, Processing Speed, and Working Memory, ranging from 44.0 to 117.0.

Autism Brain Imaging Data Exchange (ABIDE). The dataset includes anonymous resting-state functional magnetic resonance imaging (rs-fMRI) data from 17 international sites [19]. It includes brain networks from 1009 subjects, with a majority of 516 (51.14%) being patients diagnosed with Autism Spectrum Disorder (ASD). The task is to perform the binary classification for ASD diagnosis. The region definition is based on Craddock 200 atlas [42]. Given the blood-oxygen-level-dependent (BOLD) signal length of the samples in this dataset is 100, which reflects whether neurons are active or reactive, we randomly select 100 nodes to satisfy the necessary condition discussed in Proposition 5.3.2 for SPD matrices.

Philadelphia Neuroimaging Cohort (PNC). The dataset is a collaborative project from the Brain Behavior Laboratory at the University of Pennsylvania and the Children's Hospital of Philadelphia. It includes a population-based sample of individuals aged 8–21 years [159]. After the quality control, 503 subjects were included in our analysis. Among these subjects, 289 (57.46%) are female. In the resulting data, each sample contains 264 nodes with time series data collected through 120 timesteps. Hence, we randomly select 120 nodes to satisfy the necessary condition mentioned in Proposition 5.3.2 for treating generated correlation matrices as SPD. BioGender Prediction is used as the downstream task.

TCGA-Cancer Transcriptome. The Cancer Genome Atlas (TCGA) dataset is a large-scale collection of multi-omics data from over 20,000 primary cancer and matched normal bio-samples spanning 33 cancer types. In this study, we select non-

redundant cancer subjects with gene expression data and valid clinical information. The gene expression data is normalized, and the top 50 highly variable genes (HVG) are selected as the nodes for network construction. The subjects are then assigned to different samples based on their cancer subtype. The final dataset consists of 459 subjects from 66 cancer subtypes. We extract 240 correlation matrices from these subjects with 24 cancer types, each type includes ten samples, and each sample contains 50 nodes. The downstream task of this study is to predict cancer subtypes based on the HVG expression network.

Metrics

For binary classification tasks on datasets ABCD-BioGender, PNC, and ABIDE, we adopt AUROC and accuracy for a fair performance comparison. The classification threshold is set as 0.5. For the regression task on ABCD-Cog, the mean square error (MSE) is used to reflect model performance. For the multiple class classification task on TCGA-Cancer, since it contains 24 classes and each class has a balanced sample size, we take the macro Precision and macro Recall so that all classes are treated equally to reflect the overall performance. All the reported results are based on the average of five runs using different random seeds.

Table 5.3: Overall performance comparison based on the Transformer backbone. The best results are in bold, and the second best results are underlined. The ↑ indicates a higher metric value is better and ↓ indicates a lower one is better.

Method	**ABCD-BioGender**		**ABCD-Cog**	**PNC**		**ABIDE**		**TCGA-Cancer**	
	AUROC↑	Accuracy↑	MSE↓	AUROC↑	Accuracy↑	AUROC↑	Accuracy↑	Precision↑	Recall↑
w/o Mixup	95.28±0.32	87.68±1.31	60.21±1.53	74.85±4.93	66.57±6.29	73.32±4.11	66.00±3.66	35.33±11.52	45.00±10.79
V-Mixup	95.85±0.63	87.86±1.45	60.43±2.67	76.02±2.54	65.88±7.89	**75.03±5.04**	66.80±5.40	69.58±9.39	77.50±6.97
D-Mixup	94.55±2.84	87.17±3.45	60.96±1.82	76.15±4.58	68.82±6.29	72.92±4.93	67.40±5.64	70.28±12.30	75.83±12.98
DropNode	95.65±0.35	88.07±0.76	65.35±2.97	75.47±4.27	67.45±4.35	73.49±4.09	66.00±3.16	53.96±11.34	61.67±10.79
DropEdge	95.28±0.39	87.54±0.60	76.44±1.82	72.89±5.70	66.27±5.31	70.68±6.14	64.20±5.12	67.57±5.14	75.00±5.10
G-Mixup	95.24±0.92	88.16±0.63	62.16±2.04	76.01±3.04	69.41±3.21	73.68±5.67	65.60±4.56	59.72±7.77	69.44±6.27
C-Mixup	96.01±0.48	88.40±1.44	59.68±1.15	75.29±2.52	69.02±5.48	74.69±4.40	66.40±3.36	67.50±6.90	76.67±6.32
R-Mixup	**96.20±0.33**	**89.44±1.06**	**56.89±1.66**	**77.01±2.59**	**69.80±3.63**	74.79±4.90	**68.20±4.19**	**71.39±9.59**	**78.33±9.03**

Implementation Details

We equip the proposed R-MIXUP with two most popular deep backbone models for biological networks, Transformer [107] and GIN [195], to verify its universal effectiveness with different models. For the architecture of Transformer, the number of transformer layers is set to 2, followed by an MLP function to make the prediction. For each transformer layer, the hidden dimension is set to be the same as the number of nodes n, and the number of heads is set to 4. Regarding the GCN backbone, we set the number of GCN layers as 3. The graph representation is obtained with a sum readout function to make the final prediction. We randomly select 70% of the datasets for training, 10% for validation, and the remaining for testing. In the training process, we use the Adam optimizer with an initial learning rate of 10^{-4} and a weight decay of 10^{-4}. The batch size is set as 16. All the models are trained for 200 epochs, and the epoch with the best performance on the validation set is selected for the final report.

Baselines

We include a variety of Mixup approaches as baselines. Given $\Lambda \in [0,1]^{v\times v}$, $\alpha \in (0,\infty)$, $\pi \in (0,1)$, $\cdot$ is the dot product.

V-Mixup [217] is the vanilla Mixup by the linear combination of two random samples,

$$\begin{aligned} &\tilde{S} = (1-\lambda)S_i + \lambda S_j, \tilde{y} = (1-\lambda)y_i + \lambda y_j, \\ &\lambda \sim \text{Beta}(\alpha,\alpha). \end{aligned} \tag{5.15}$$

D-Mixup is the discrete Mixup, a naive baseline designed by ourselves. Given two randomly selected samples, a synthetic sample is generated by obtaining parts of the

edges from one sample and the rest from the other,

$$\begin{aligned} \tilde{S} = (1-\Lambda)\cdot S_i + \Lambda \cdot S_j, \tilde{y} = (1-\lambda)y_i + \lambda y_j, \\ \Lambda^{i,j} \sim \mathrm{B}(\lambda), \lambda \sim \mathrm{Beta}(\alpha, \alpha). \end{aligned} \tag{5.16}$$

DropNode [81] randomly selects nodes given a sample and sets all edge weights related to these selected nodes as zero,

$$\tilde{S} = \Lambda \cdot S, \Lambda^{p,:} = \Lambda^{:,p} = z, z \sim \mathrm{Bernoulli}(\pi). \tag{5.17}$$

DropEdge [155] randomly selects edges given a sample and assigns their weights as zero,

$$\tilde{S} = \Lambda \cdot S, \Lambda^{p,q} \sim \mathrm{Bernoulli}(\pi). \tag{5.18}$$

G-Mixup [82] is originally proposed for classification tasks, which augments graphs by interpolating the generator of different classes of graphs. Since each cell in a covariance and correlation matrix represents a specific edge in a graph, we can convert a graph generator into a group of generator for each edge. We model each edge generator as a conditional multivariate normal distribution $P(S^{p,q} \mid y)$. The augmentation process can be formulated as,

$$\begin{aligned} \tilde{S}^{p,q} \sim (1-\lambda)P(S^{p,q} \mid y_i) + \lambda P(S^{p,q} \mid y_j), \tilde{y} = (1-\lambda)y_i + \lambda y_j, \\ \lambda \sim \mathrm{Beta}(\alpha, \alpha). \end{aligned} \tag{5.19}$$

For the setting of classification,

$$P(S^{p,q} \mid y = c) \sim \mathcal{N}\left(\mu_c^{p,q}, (\sigma_c^{p,q})^2\right), \tag{5.20}$$

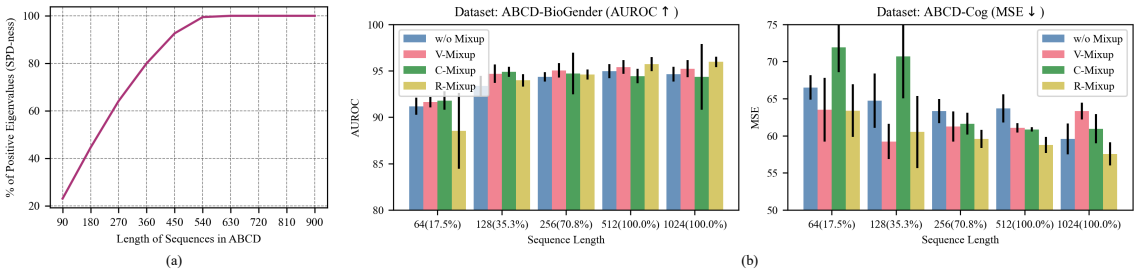


Figure 5.4: (a) The influence of time-series sequence length t on the percentage of the positive eigenvalues (%). (b) The influence of the sequence length t or *SPD-ness* (%) on the prediction performance of classification and regression tasks.

To extend G-Mixup for regression, we slightly modify the augmentation process to adapt it for regression tasks as

$$P(S^{p,q} \mid y) \sim \mathcal{N}\left(\mu^{p,q} + \frac{\sigma^{p,q}}{\sigma_y}\rho^{p,q}\left(y - \mu_y\right),\ \left(1 - (\rho^{p,q})^2\right)(\sigma^{p,q})^2\right), \tag{5.21}$$

where μ and σ are the mean and standard deviation of the weight for each edge, ρ is the correlation coefficient between $S^{p,q}$ and y.

C-Mixup [205] shares the same process with the V-Mixup. Instead of randomly selecting two samples, C-Mixup picks samples based on label distance to ensure the mixed pairs are more likely to share similar labels $(S_j, y_j) \sim P\left(\cdot \mid (S_i, y_i)\right)$, where P is a sampling function which can sample closer pairs of examples with higher probability. For classification tasks, it degenerates into the intra-class V-Mixup.

Table 5.4: Detailed performance comparison of different sample sizes with Transformer as the backbone.

Percentage (in %)	Dataset: ABCD-BioGender (AUROC↑)				Dataset: ABCD-Cog (MSE↓)			
	w/o Mixup	V-Mixup	C-Mixup	R-Mixup	w/o Mixup	V-Mixup	C-Mixup	R-Mixup
10	87.14±1.15	88.99±0.75	88.72±1.13	**90.21±0.64**	73.07±2.75	77.00±4.58	71.22±1.68	**70.69±1.06**
20	90.60±0.91	91.11±0.54	91.49±0.89	**92.72±0.64**	69.70±2.75	69.80±2.42	69.30±3.21	**66.50±2.50**
30	92.60±0.51	93.45±0.35	93.33±0.78	**93.93±0.55**	65.97±2.48	65.84±1.11	64.31±0.57	**63.50±1.61**
40	92.84±0.40	94.06±0.48	93.95±0.53	**94.12±0.21**	63.91±4.07	63.14±1.08	61.88±2.93	**61.15±1.80**
50	94.18±0.51	**95.20±0.39**	95.03±0.57	94.78±0.98	61.89±3.85	63.45±1.65	61.26±1.31	**60.82±2.71**
60	94.22±0.44	95.19±0.54	95.17±0.32	**95.65±0.37**	59.47±1.59	60.32±0.94	60.20±1.58	**58.75±1.65**
70	94.18±0.40	**95.51±0.18**	95.49±0.28	95.07±0.18	62.35±2.28	61.15±1.51	60.54±3.57	**60.17±0.50**
80	95.18±0.31	95.60±0.42	95.73±0.51	**95.94±0.31**	59.85±1.47	60.31±1.07	60.85±3.84	**56.78±2.05**
90	95.55±0.86	**95.92±0.34**	95.49±0.73	95.24±0.65	61.17±3.36	61.51±0.78	60.35±0.93	**57.45±3.39**
100	95.28±0.32	95.85±0.63	96.01±0.48	**96.20±0.33**	60.21±1.53	60.43±2.67	59.68±1.15	**56.89±1.66**

5.4.2 RQ1: Performance Comparison

Overall Performance. The overall comparison based on the Transformer and GCN backbone are presented in Table 5.3 and Table C.1 respectively, where *ABCD-BioGender*, *PNC*, *ABIDE*, and *TCGA-Cancer* focus on classification tasks, while *ABCD-Cog* is a regression task. Since the performance of the two backbones demonstrates similar patterns, we focus on the result discussion of the Transformer due to the space limit. Specifically, for classification tasks, incorporating the Mixup technique can constantly improve the performance, especially on the *TCGA-Cancer* dataset, which features a small sample size with high dimensional matrices. Among the various Mixup techniques, our proposed R-Mixup performs the best across datasets and tasks, indicating the further advantage of using log-Euclidean metrics instead of Euclidean metrics for SPD matrices mixture. Besides, for datasets with a relatively smaller sample size, such as PNC, ABIDE, and TCGA-Cancer, R-Mixup can further reduce training variance and stabilize the final performance compared with other data augmentation methods.

Compared with the improvements on classification tasks, R-Mixup demonstrates a more significant advantage on the regression task. It is shown that R-Mixup can significantly reduce the MSE compared with the baseline without Mixup (5.5% with the transformer backbone) and archive a large advantage over the second runner (4.8% with the transformer backbone). It is also noted that other Mixup approaches sometimes hurt the model performance, indicating the Euclidean space cannot measure the distance between SPD matrices very well, and the mixed samples may not be paired with the correct labels. In contrast, our proposed log-Euclidean metric can correctly represent the distance among SPD matrices and therefore address the problem of *arbitrarily incorrect label*.

Performance with Different Sample Sizes. As collecting labeled data can be extremely expensive for biological networks in practice, we adopt R-Mixup for the

challenging low-resource setting to justify its efficacy with limited labeled data only. For this set of experiments, we vary the training sample size from 10% to 100% of the full datasets to show the performance of R-Mixup based on transformers with different sample sizes. Specifically, the ABCD dataset is adopted in this detailed analysis due to its relatively large sample size and supports for both classification and regression tasks. The selected comparing methods are the strongest baselines, namely V-Mixup and C-Mixup, from the overall performance in Table 5.3. Results are presented in Table 5.4.

On the classification task of BioGender prediction, impressively, the proposed R-Mixup can already achieve a decent performance with only 10% percent of full datasets and demonstrates a large margin over other compared methods. As the sample size becomes larger, the performance of different data augmentation methods tends to be close, while the proposed R-Mixup reaches the best performance for most of the cases (7 out of 10 setups). On the more challenging regression task of Cognition Summary Score prediction, R-Mixup consistently outperforms the other two baselines under different portions of the training data, which stresses the absolute advantages of our proposed R-Mixup in its flexible and effective adaption for the regression settings. Note that when equipped with an inappropriate augmentation method (i.e., V-Mixup), the regression performance can always deteriorate under different volumes of training data. This implies the necessity of proposing appropriate Mixup techniques tailored for biological networks to address specific challenges for regression tasks. Furthermore, we propose a case study in Appendix C.7 to show why R-Mixup can achieve the best performance for the Regression task in the ABCD-Cog dataset.

5.4.3 RQ2: The Relations of Sequence Length, SPD-ness and Model Performance

To quantitatively verify the necessary conditions of SPD matrices in Proposition 5.3.2, we vary the length of sequences whose pairwise correlations compose the network matrices and observe its influence on the percentage of positive eigenvalues and the final prediction performance. For better illustration, we define a new terminology *SPD-ness* to reflect the percentage of positive values among all eigenvalues. The higher the percentage of positive eigenvalues, the higher *SPD-ness*, and a full SPD matrice requires all the eigenvalues to be positive. Specifically, we choose the dataset with the longest time sequence, namely ABCD, to facilitate this study. Since samples in the ABCD dataset are of different sequence lengths, we simply select those with sequence length longer than 1024 and truncate them to 1024 to form a length-unified dataset *ABCD-1024*, leading to 4613 samples for the *ABCD-BioGender* classification task and 4533 samples for the *ABCD-Cog* regression task.

First, we investigate the relationship between the length of biological sequences t and the *SPD-ness* of the corresponding network matrix. The results are shown in Figure 5.4(a), where the value of sequence length t is varied from 90 to 900 with a step size of 90. For each given t, we construct the correlation matrices based on each pair of the truncated sequences with only the first t elements from the original sequences. Then the eigenvalue decomposition is applied to each obtained correlation matrix, and the percentage of positive ($> 10^{-6}$) eigenvalues are calculated. The reported results are the average over all the correlation matrices. From this curve, we observe that the percentage of positive eigenvalues grows gradually as the time-series length increases. The growth trend gradually slows down, reaching a percentage point saturation at about the length of 540, where the full percentage indicates full *SPD-ness*. Note that the number of variables n for the ABCD dataset is 360. This aligns with our conclusion in Proposition 5.3.2 that a necessary condition for correlation matrices

satisfying SPD matrices is $t \geq n$.

Second, with the verified relation between the sequence length t and *SPD-ness*, we study the influence of sequence length t or *SPD-ness* on the prediction performance. We observe that directly truncating the time series to length t will lose a huge amount of task-relevant signals, resulting in a significant prediction performance drop. As an alternative, we reduce the original sequence to length t by taking the average of each $1024/t$ consecutive sequence unit. Results on the classification task *ABCD-BioGender* and the regression task *ABCD-Cognition* with the input of different time-series length t are demonstrated in Figure 5.4(b). It shows that for the classification task, although V-Mixup and C-Mixup demonstrate an advantage when the percentage of positive eigenvalues is low, the performance of the proposed R-MIXUP continuously improves as the sequence length t increases and finally beats the other baselines. For the regression task, our proposed R-MIXUP consistently performs the best regardless of the *SPD-ness* of the correlation matrices. The gain is more observed when the dataset matrices are full SPD. Combining these observations from both classification and regression tasks, we prove that the proposed R-MIXUP demonstrates superior advantages for mixing up SPD matrices and facilitating biological network analysis that satisfies full *SPD-ness*.

5.4.4 RQ3: Hyperparameter and Efficiency Study

The Influence of Key Hyperparameter α. We study the influence of the key hyperparameter α in R-MIXUP, which correspondingly changes the Beta distribution of λ in Equation (5.13). Specifically, the value of α is adjusted from 0.1 to 1.0, and the corresponding prediction performance under the specific values is demonstrated in Figure 5.5. We observe that the prediction performance of both classification and regression tasks are relatively stable as the value of α varies, indicating that the proposed R-MIXUP is not sensitive to the key hyperparameter α.

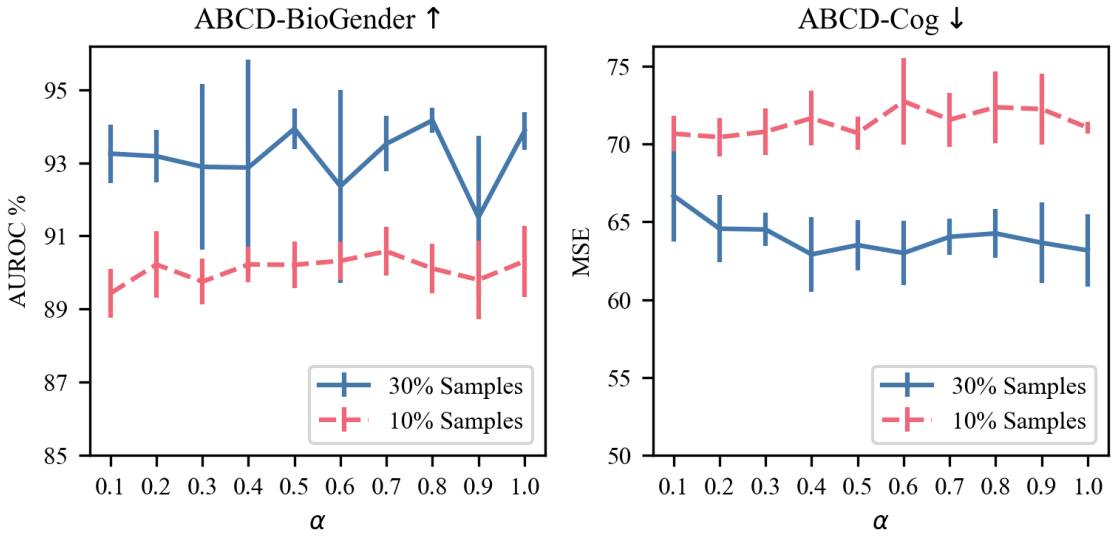


Figure 5.5: The influence of the key hyperparameter (α) value on the performance of classification and regression tasks.

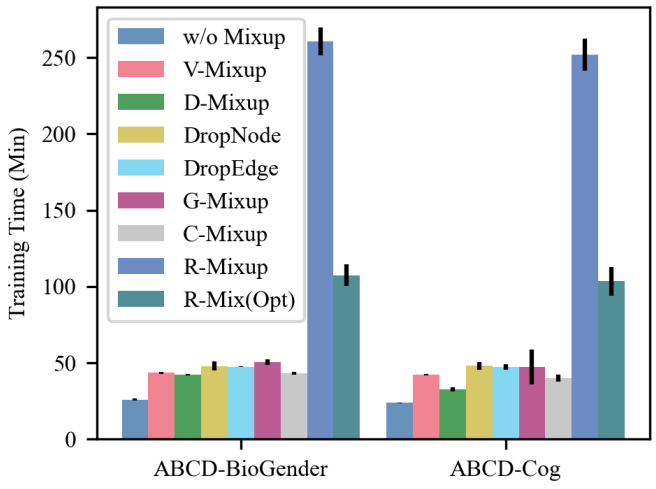


Figure 5.6: Training Time of different Mixup methods on the large ABCD dataset. R-Mixup is the original model while R-Mix(Opt) is time-optimized as discussed in Section 5.3.5.

Efficiency Study. To further investigate the efficiency of different Mixup methods, we compare the training time of different data augmentation methods on the large-scale dataset, ABCD, to highlight the difference. The results are shown in Figure 5.6. Besides, the running time comparison on three smaller datasets, ABIDE, PNC, and TCGA-Cancer are also included in appendix C.4 for reference. All the compared methods are trained with the same backbone model [107]. It is observed that with the precomputed eigenvalue decomposition, the training speed of the optimized R-Mixup on the large ABCD dataset can be 2.5 times faster than the original model without optimization. Besides, on the smaller datasets such as PNC, ABIDE, and TCGA-Cancer, there is no significant difference in elapsed time between different methods.

5.5 Conclusion

In this paper, we present R-Mixup, an effective data augmentation method tailored for biological networks that leverage the log-Euclidean distance metrics from the Riemannian manifold. We further propose an optimized strategy to improve the training efficiency of R-Mixup. Empirical results on five real-world biological network datasets spanning both classification and regression tasks demonstrate the superior performance of R-Mixup over existing commonly used data augmentation methods under various data scales and downstream applications. Besides, we theoretically verify a necessary condition overlooked by prior works to determine whether a correlation matrix is SPD and empirically demonstrate how it affects the prediction performance, which we expect to guide future applications spreading the biological networks.

Chapter 6

Multi-Task Learning Framework: Leveraging Diverse Prediction Targets for Enhanced Individual Task Performance and Dataset Utilization

6.1 Introduction

Adolescent Brain Cognitive Development (ABCD) study [22] is the largest and most long-term study of brain development and child health in the US. It provides a vast brain development dataset in a diverse population, including functional magnetic resonance imaging data (fMRI) and abundant biological and behavioral survey results. This dataset offers an opportunity to explore the relationship between intricate brain connections and various behavioral data [113, 30, 76].

Leveraging the potential of neuroimaging data, recent studies have shown a grow-

ing trend of using brain networks derived from fMRI to predict various clinical outcomes and individual behaviors with different models [114, 111, 126, 44, 46]. Researchers have also developed innovative approaches to analyze these models and uncover potential correlations between functional brain networks and predicted outcomes. For example, Kawahara et al.[111] introduced BrainNetCNN, a convolutional neural network designed to predict cognitive and motor developmental outcome scores from brain networks. Similarly, Li et al.[126] proposed a graph neural network model to predict clinical targets and discovered task-specific neurological biomarkers, demonstrating the effectiveness of graph-based approaches in capturing meaningful patterns in brain networks. Chen et al. [27] further extended this line of research by training kernel regression models for 36 tasks and analyzing task relationships based on the learned model. These studies highlight the potential of leveraging brain networks to gain insights into the underlying neural mechanisms associated with various clinical outcomes and individual behaviors.

Multi-task learning (MTL)[43, 162, 141, 8] has emerged as a promising approach for improving the generalization abilities of predictive models by enabling multiple learning tasks to share their knowledge. In the context of brain network analysis in ABCD, where there is a diverse range of prediction targets, MTL can be particularly beneficial. By training several tasks simultaneously, MTL allows for a more native capture of task correlations, potentially leading to improved individual task performance. This contrasts with the approach taken by Chen et al.[27], which builds individual models for each behavior task. By leveraging the shared representations learned across tasks, MTL can enhance the model's ability to uncover the underlying relationships between brain networks and various behavioral measures, resulting in more accurate and generalizable predictions.

In this work, we propose a novel MTL framework that jointly trains 35 tasks using multi-view functional brain networks from 6,682 samples in the ABCD study. We

employ the Brain Network Transformer [107] as the backbone model, which converts a brain network into a graph-level embedding. This embedding is then fed into task-specific fully connected networks (FCNs) for each prediction target. By learning shared representations across tasks while still allowing for task-specific predictions, our approach aims to leverage the commonalities between tasks and improve overall prediction performance. Our main contributions are summarized as follows:

- We propose a novel MTL framework for predicting various measures from multi-view brain network data using a graph transformer architecture. Our approach learns shared representations across tasks while allowing for task-specific predictions, improving performance compared to single-task learning. Besides, the ablation study shows the effectiveness of two key training strategies during Multi-Task training.
- We conduct extensive experiments on the ABCD dataset, including 35 tasks categorized into three domains: *cognition*, *personality*, and *mental health*. We demonstrate the impact of MTL on different types of tasks.
- We develop innovative visualization techniques based on integrated gradients to interpret the learned task correlations and identify influential brain network edges, contributing to a better understanding of the complex relationships between brain structure and behavioral outcomes.

6.2 Method

6.2.1 Problem Definition

Let $\mathcal{D} = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{n}$ be a dataset consisting of n samples. For each sample i, the input $\mathbf{X}^{(i)} = \{X_1^{(i)}, X_2^{(i)}, \ldots, X_v^{(i)}\}$ represents a collection of v brain networks, each derived from a distinct fMRI task (e.g., resting-state, stop-signal task, and N-Back). These brain networks, denoted by $X_j^{(i)} \in \mathbb{R}^{M \times M}$, capture the func-

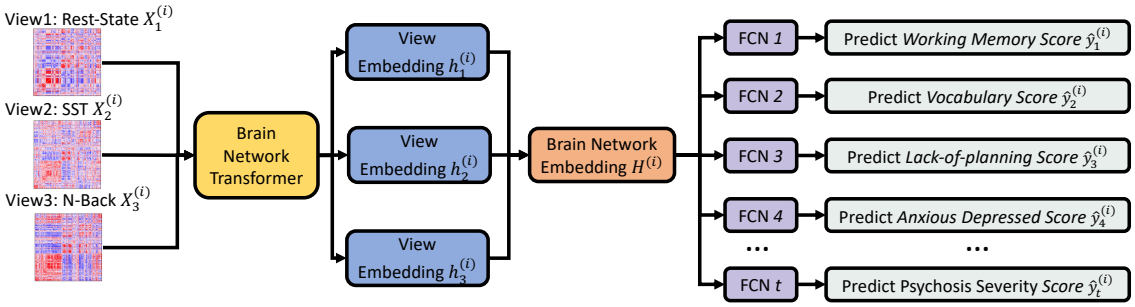


Figure 6.1: Overview of our multi-task learning framework for predicting various measures from multi-view brain networks. Given a set of brain networks $\{X_1^{(i)}, X_2^{(i)}, \ldots, X_v^{(i)}\}$ derived from different views for the i-th subject, the Brain Network Transformer generates a unified brain network embedding $H^{(i)}$. This embedding is then fed into task-specific FCN to predict the corresponding target scores $\{\hat{y}_1^{(i)}, \hat{y}_2^{(i)}, \ldots, \hat{y}_t^{(i)}\}$ for various measures. The entire framework is trained end-to-end using multi-task learning, allowing for the sharing of knowledge across tasks while still enabling task-specific predictions.

tional connectivity between M brain regions. The corresponding prediction target $\mathbf{Y}^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \ldots, y_t^{(i)}\}$ is a set of t behavioral measures, such as cognitive scores, personality traits, or mental health indicators, associated with the i-th subject. In short, given this multi-view, multi-task dataset, our goal is to develop a predictive model that leverages the complementary information from the v brain networks to simultaneously predict the t behavioral outcomes.

6.2.2 Model Architecture

Fig. 6.1 shows the proposed multi-task learning framework for predicting behavioral outcomes from multi-view brain networks.

Shared Representation Learning. The footstone of our framework is Brain Network Transformer (BNT) [107], which serves as the shared backbone model. BNT is designed to process individual views of brain networks, denoted as $X_j^{(i)}$, where $j \in \{1, \ldots, v\}$ indexes the view and $i \in \{1, \ldots, n\}$ indexes the sample. For each view j, BNT learns a hidden representation embedding $h_j^{(i)} = \text{BNT}(X_j^{(i)})$. These view-specific embeddings capture the patterns present in the corresponding brain

networks derived from distinct fMRI tasks. To obtain a comprehensive representation of each sample, we concatenate the view-specific embeddings $h_j^{(i)}$ from all v views, resulting in a sample-level embedding $H^{(i)} = \bigoplus_{j=1}^{v} h_j^{(i)}$, where $\bigoplus$ denotes the concatenation operation. This sample-level embedding integrates the multifaceted information captured across different fMRI views, providing a holistic representation of each individual's brain connectivity patterns.

Task-specific Prediction. To achieve multi-task learning, we employ a separate Multi-Layer Perceptron (MLP) for each task $k \in \{1, \ldots, t\}$. These task-specific MLPs take the sample-level embedding $H^{(i)}$ as input and predict the corresponding behavioral outcome $\hat{y}_k^{(i)} = \text{MLP}_k(H^{(i)})$. By leveraging dedicated MLPs for each task, our framework allows for task-specific adaptations while benefiting from the shared representation learned by the BNT.

6.2.3 Multi-task Training Strategies

The entire framework is trained end-to-end using a multi-task learning approach. However, training a model to simultaneously predict multiple behavioral outcomes presents several challenges due to the diverse characteristics and varying scales of the prediction targets. We introduce two key strategies to address these challenges and ensure effective training: Batch-Wise Loss Balancing and Target Standardization. Our ablation study results in Section 6.3 show the necessity of incorporating them during training.

Batch-Wise Loss Balancing. In multi-task learning, tasks with larger loss values can potentially dominate the training process, hindering the model's ability to learn from all tasks equally. To mitigate this issue, we employ a batch-wise loss balancing technique that adaptively adjusts the weight of each task's loss within a training batch. Let L_k denote the loss associated with task k, where $k \in \{1, \ldots, t\}$. We compute the balanced loss $\hat{L}_k$ for each task as follows: $\hat{L}_k = \frac{L_k}{\bar{L}_k}$, where $\bar{L}_k$ is the

average loss value for task k over the current batch of samples. By normalizing each task's loss to have an average value of 1, we ensure that all tasks contribute equally to the overall optimization process. The total balanced loss L_{total} is then calculated as the sum of all individual balanced task losses: $L_{\text{total}} = \sum_{k=1}^{t} \hat{L}_k$.

Target Standardization. Another challenge in multi-task learning is the varying scales of the target variables across different tasks. To address this issue, we employ a target standardization preprocessing step. For a regression task k, where $k \in \{1, \ldots, t\}$, we standardize the training labels $y_k^{(i)}$ to have zero mean and unit variance: $y_k^{(i)} = \frac{y_k^{(i)} - \mu_k}{\sigma_k}$, where μ_k and σ_k denote the mean and standard deviation of the training labels for task k, respectively. This normalization brings all tasks to a similar scale, facilitating the model's ability to learn from them concurrently. During the validation and testing phase, we apply the inverse of this normalization process to transform the predicted labels $\hat{y}_k^{(i)}$ back to their original scale. Using the mean μ_k and standard deviation σ_k computed from the training set, we perform the following operation: $\hat{y}_k^{(i)} = \hat{y}_k^{(i)} \cdot \sigma_k + \mu_k$. By standardizing the targets during training and reversing the normalization during inference, we ensure that the model can effectively learn from tasks with different scales while producing predictions in the original target domain.

6.3 Experiments

Dataset. We use the Adolescent Brain Cognitive Development (ABCD) dataset [22], which includes fMRI and behavioral data from a large cohort of children. We utilize resting-state and task-based fMRIs (stop-signal task [133] and N-Back task [37]) for brain network construction based on the HCP 360 ROI atlas [74]. In this study, we aim to predict 35 distinct labels, which span across 15 *neurocognitive ability* [134], 9 *impulsivity-related personality* and 11 *mental health assessments* [10], as detailed in Supplementary D.1. The experimental dataset includes 6,682 samples with quality

control procedures and filtering these samples with incomplete fMRI and behavioral data.

Setting. We employ a Brain Network Transformer as the shared model backbone, which consists of 3 transformer layers with 4 attention heads and an output dimension matching the number of nodes (360) in the brain network. The transformer layers are followed by task-specific 3-layer MLP branches with activation functions, each responsible for making predictions for one of the tasks. Since all tasks are regression tasks, we use Mean Squared Error (MSE) Loss as the loss function for all tasks. The model, which can predict 35 tasks simultaneously, has a total of 22.52 million parameters. We randomly split the ABCD dataset into training (70%), validation (10%), and testing (20%) subsets. During training, we use the Adam optimizer with a weight decay of 10^{-4}, a cosine learning rate scheduler (initial: 10^{-4}, final: 10^{-5}), and a batch size of 16. The model is trained for 100 epochs, and the model whose epoch shows the best total loss on the validation set is selected as the final model to report performance.

Metrics. To evaluate our multi-task learning model's performance on the 35 regression tasks from the ABCD dataset, we use two metrics: Mean Squared Error (MSE) and R-squared (R^2). MSE measures the average squared difference between predicted and actual values, with lower values indicating better performance. R^2, on the other hand, is used to compare performance across tasks with varying label scales. R^2 values range from $-\infty$ to 1, where negative values indicate worse performance than using the target variable's mean, zero indicates equivalence to using the mean, and positive values suggest the model captures useful information from brain networks and the prediction beats the mean of the target. Thus, R^2 can be used to evaluate a task's predictability. All reported results are averaged over 5 runs with different random seeds.

Performance Evaluation. The overall results of our multi-task learning model on

Table 6.1: Performance comparison of single-task, multi-task, and multi-task (Cognition tasks only) models on the ABCD dataset. Tasks within each type (e.g., Cognition, Personality) are sorted in descending order based on the R^2 value under the Single-Task column. Tasks highlighted in purple have an R^2 value greater than or equal to 0.03, indicating that they are predictable. **Bold** values indicate the best result for these predictable tasks across the three model settings. The ↑ indicates a higher metric value is better, and ↓ indicates a lower one is better.

Type	Task	Single-Task		Multi-Task		Multi-CogTask	
		MSE ↓	R^2 ↑	MSE ↓	R^2 ↑	MSE ↓	R^2 ↑
Cognition	OverallCognition	54.34±2.35	**0.26±0.02**	**53.79±2.27**	0.24±0.04	58.30±3.15	0.20±0.03
	CrystallizedCognition	**32.54±2.11**	**0.25±0.03**	33.01±0.99	0.24±0.02	35.13±2.62	0.19±0.05
	Vocabulary	47.39±2.89	0.20±0.03	**47.15±2.87**	**0.21±0.04**	49.04±2.94	0.16±0.05
	Reading	35.57±1.79	0.15±0.03	**34.17±0.83**	**0.15±0.03**	36.85±2.61	0.14±0.03
	FluidCognition	89.97±2.24	**0.14±0.03**	**87.87±4.55**	0.12±0.05	92.50±3.95	0.11±0.02
	FluidIntelligence	12.56±0.32	0.12±0.01	12.28±0.25	0.12±0.03	**12.03±0.60**	**0.13±0.02**
	WorkingMemory	121.76±5.44	0.07±0.03	**116.12±2.65**	**0.09±0.03**	120.12±3.06	0.09±0.03
	ExecutiveFunction	72.55±3.74	0.07±0.01	**70.33±2.33**	**0.07±0.02**	73.09±3.98	0.07±0.02
	ShortDelayRecall	8.43±0.18	0.06±0.03	**7.87±0.32**	**0.08±0.02**	8.16±0.32	0.06±0.02
	LongDelayRecall	9.07±0.33	0.05±0.02	**8.62±0.24**	**0.07±0.03**	9.14±0.22	0.04±0.02
	VisuospatialAccuracy	0.03±0.00	0.05±0.04	**0.03±0.00**	**0.09±0.02**	0.03±0.00	0.08±0.01
	Attention	72.84±4.41	0.04±0.01	71.99±3.30	0.04±0.04	**70.20±2.74**	**0.05±0.02**
	EpisodicMemory	136.82±4.87	0.04±0.02	**136.50±5.84**	**0.04±0.02**	138.15±3.91	0.04±0.03
	ProcessingSpeed	**198.17±5.99**	**0.03±0.01**	200.26±8.89	0.02±0.02	203.91±8.01	0.00±0.03
	VisuospatialReactionTime	208k±7k	0.00±0.00	212k±9k	0.00±0.00	210k±4k	0.01±0.01
Personality	RewardResponsiveness	8.44±0.37	0.01±0.00	8.33±0.51	-0.02±0.05	-	-
	Drive	8.61±0.20	0.01±0.01	8.66±0.47	0.01±0.04	-	-
	PositiveUrgency	8.02±0.22	0.01±0.01	8.17±0.27	-0.00±0.06	-	-
	LackOfPlanning	5.09±0.16	0.00±0.01	5.32±0.22	-0.01±0.01	-	-
	LackPerseverance	4.86±0.11	0.00±0.01	4.54±0.19	-0.01±0.01	-	-
	FunSeeking	6.62±0.16	0.00±0.00	6.72±0.22	-0.01±0.04	-	-
	SensationSeeking	7.14±0.15	-0.00±0.00	7.10±0.19	-0.01±0.02	-	-
	BehavioralInhibition	13.48±0.10	-0.01±0.01	13.50±0.71	-0.02±0.03	-	-
	NegativeUrgency	6.90±0.18	-0.01±0.03	6.85±0.39	-0.02±0.07	-	-
Mental Health	TotalPsychosisSymptoms	11.21±0.69	0.01±0.01	10.76±0.90	-0.01±0.04	-	-
	AttentionProblems	10.54±0.55	0.00±0.02	10.80±0.82	-0.00±0.04	-	-
	AnxiousDepressed	8.68±0.59	-0.00±0.00	8.67±0.84	-0.04±0.06	-	-
	AggressiveBehavior	16.93±1.66	-0.00±0.01	16.45±1.46	-0.02±0.06	-	-
	WithdrawnDepressed	2.85±0.13	-0.00±0.01	2.57±0.18	-0.03±0.05	-	-
	SomaticComplaints	3.80±0.19	-0.00±0.01	3.67±0.33	-0.02±0.04	-	-
	ThoughtProblems	4.53±0.19	-0.00±0.00	4.31±0.39	-0.03±0.05	-	-
	SocialProblems	4.66±0.24	-0.01±0.01	4.36±0.47	-0.01±0.07	-	-
	PsychosisSeverity	87.75±7.75	-0.01±0.04	89.23±8.65	-0.00±0.04	-	-
	Mania	6.50±0.51	-0.01±0.01	6.49±0.93	-0.01±0.05	-	-
	RuleBreakingBehavior	3.01±0.26	-0.02±0.03	2.75±0.15	0.01±0.04	-	-

the ABCD dataset are shown in Table 6.1. From the table, we can obtain 3 key insights: (1) Single-task performance: The **Single-Task** column reveals that for all Personality and Mental Health tasks, these models' R^2 is below 0.03, indicating that there is limited predictive power when using brain networks to predict these labels. In contrast, for the Cognition tasks, except for the Visuospatial Reaction Time task, all other **14** tasks have an R^2 greater than or equal to 0.03. This suggests that the model can capture useful information from the brain networks and outperform pre-

dictions based solely on the mean of the target variable for these Cognition tasks; (2) Multi-task learning benefits: By comparing the Single-Task performance column with the **Multi-Task** performance column, we observe that multi-task training improves the performance of almost all tasks that already exhibit predictive power in the single-task setting. However, the Personality and Mental Health tasks that were unpredictable in the single-task setting remain unpredictable in the multi-task setting, indicating that these tasks cannot effectively leverage useful information from other tasks during joint training; (3) Impact of removing unpredictable tasks: To further investigate the influence of the unpredictable *Personality* and *Mental Health* tasks on the overall model performance, we conducted an additional experiment where we removed these tasks during multi-task training. The results of this experiment are shown in the **Multi-CogTask** column. Interestingly, we observe that by excluding these unpredictable tasks, the performance of the remaining tasks drops compared to the multi-task setting that includes all tasks. This finding suggests that labeling information from Personality and Mental Health tasks is still helpful for other tasks, even though these tasks themselves remain unpredictable.

Ablation Study. We investigate the effectiveness of our two key training strategies in our multi-task learning model: *Batch-Wise Loss Balancing* and *Target Standardization*. We compare the performance of our full model with three ablated versions: (1) without Batch-Wise Loss Balancing, (2) without Target Standardization, and (3) without both strategies. The results in Fig. 6.2 show that removing Batch-Wise Loss Balancing leads to a slight decrease in performance across all 14 predictable tasks, while removing Target Standardization causes a significant drop. When both strategies are removed, the model fails to learn any meaningful information, resulting in negative $\mathbf{R^2}$ values for all tasks. This study demonstrates the importance of these training strategies in enabling successful multi-task learning for brain network analysis.

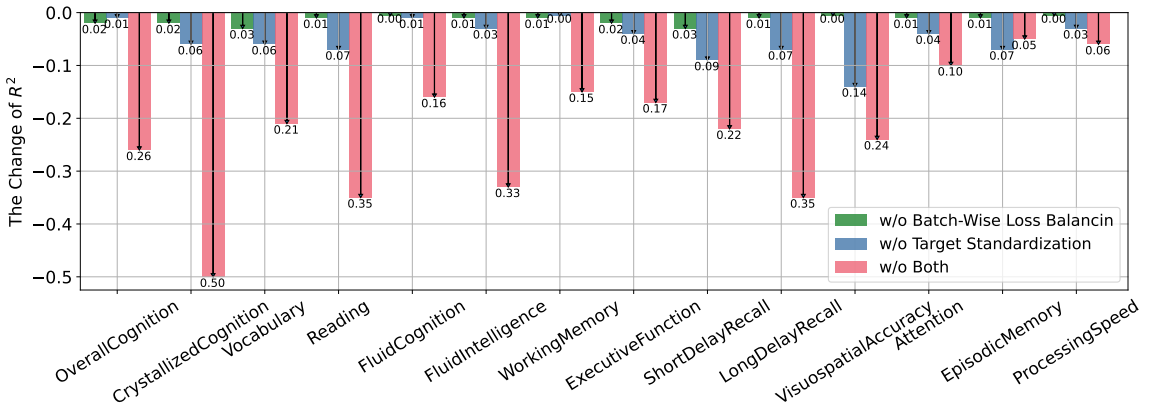


Figure 6.2: Ablation study results comparing the performance of our full multi-task learning model with three ablated versions. The bars represent the difference in R^2 values between each ablated version and the full model for the 14 predictable tasks.

6.4 Task Correlation Analysis

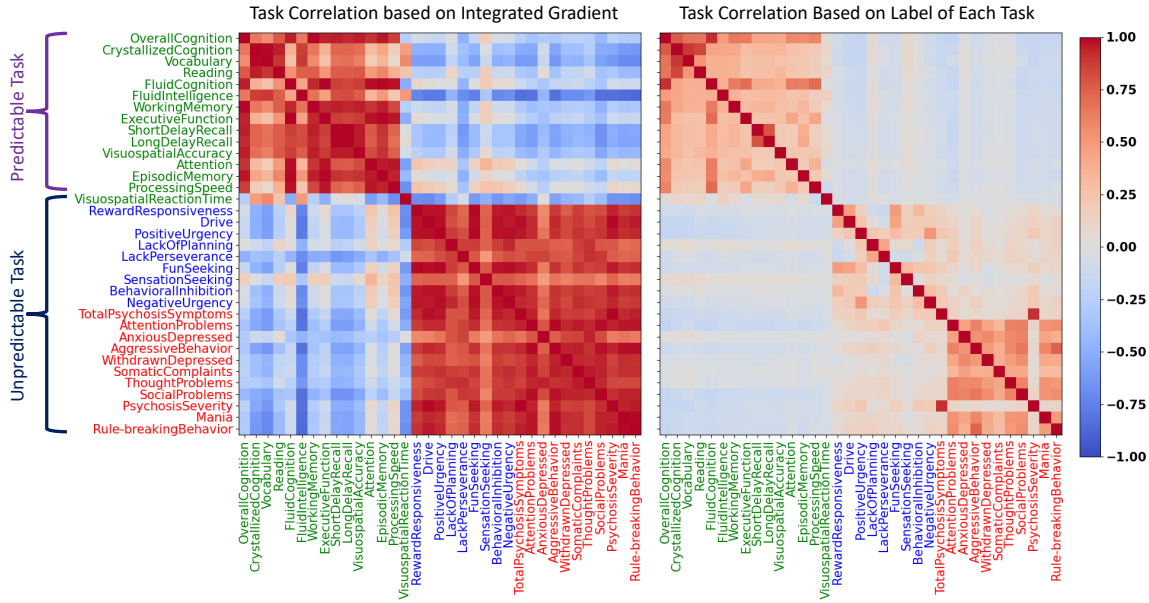


Figure 6.3: Task correlation matrices based on integrated gradients (left) and task labels (right). The integrated gradients matrix reveals the correlation among tasks regarding their importance to the model's predictions, while the label correlation matrix shows the inherent relationships among task labels. The task names are color-coded based on their type: green for cognition, blue for personality, and red for mental health. The comparison of these two matrices provides insights into the model's ability to capture meaningful task relationships from data.

In this section, we visualize the task correlation matrix learned by our multi-task

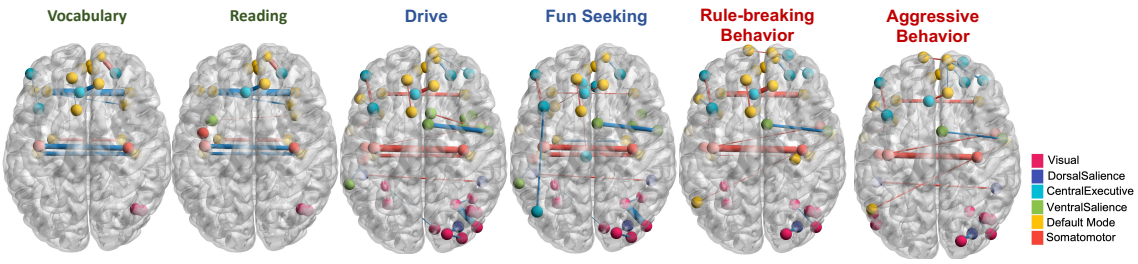


Figure 6.4: Visualization of the top 0.05% brain network edges for 6 tasks, determined by integrated gradients $\mathbf{G}^k$. Node color indicates functional module, while edge color (blue for negative, red for positive) and thickness represent integrated gradient magnitude. This figure reveals key brain edges the model relies on for predictions in each task.

learning model (Fig. 6.3 Left) and compare it with the inherent correlations among task labels (Fig. 6.3 Right). To achieve this, we employ the integrated gradients method [175], which allows us to track the importance of each edge in brain networks contributing to the model's prediction. The process to obtain the task-level correlation matrix C and edge importance $\mathbf{G}^k$ for each task k is described in Supplementary D.2. Fig. 6.3 Left reveals two task groups: a. the 14 most predictable tasks and b. all personality and mental health tasks, with strong positive correlations within each group. Besides, the first group can also be found in the task label correlation (Fig. 6.3 Right), which shows our model can capture meaningful task relationships from data. Fig. 6.4 visualizes the top 0.05% edges with the highest absolute importance values from G^k for each task k, demonstrating similar patterns of important edges within the same task group.

6.5 Conclusion

We proposed a novel multiple-task learning framework for predicting cognitive, personality, and mental health measures from brain networks using the ABCD dataset. Our approach effectively captures meaningful relationships across tasks and improves prediction performance compared to single-task learning. Through experiments, we

demonstrated the importance of two training strategies and provided deep task correlation analysis by the integrated gradient method.

Chapter 7

Conclusion

7.1 Summary of Achievements

In this dissertation, we have made significant contributions to the field of deep learning for brain network analysis by developing innovative techniques that address the key challenges of limited sample sizes, high-dimensional data, and the need for interpretable predictions. Our proposed methods span both model architectures and training strategies, pushing the boundaries of what is possible in neuroimaging research.

We introduced three novel model architectures specifically designed for brain network analysis, including FbNetGen, which generates task-aware functional brain networks from raw fMRI data; Brain Network Transformer (BNT), which captures the unique properties of brain networks using a transformer-based architecture; and Dynamic bRAin Transformer (DRAT), which models the temporal dynamics of brain networks for improved predictions and interpretability. These architectures have demonstrated superior performance on large-scale fMRI datasets and provided valuable insights into the complex relationships between brain networks and various cognitive functions and disorders.

Furthermore, we developed advanced training strategies to enhance the generalization and performance of deep learning models for brain network analysis. R-mixup, our proposed data augmentation approach, effectively addresses the limited sample size challenge by operating on the Riemannian manifold of symmetric positive definite matrices. Additionally, our multi-task learning framework enables knowledge sharing across related tasks and improves individual task performance by jointly predicting various behavioral and clinical measures from brain networks.

The extensive experiments conducted on multiple datasets and tasks showcase the practical value and superior performance of our proposed methods. By addressing the key challenges in deep learning for brain network analysis, this dissertation contributes to a better understanding of the human brain and its role in cognition and disorders, with potential applications in neuroimaging research and clinical settings.

7.2 Future Directions

The work presented in this dissertation opens up several exciting avenues for future research in deep learning for brain network analysis. We highlight two particularly promising directions:

7.2.1 Causality and Effective Connectome for Brain Network Analysis

One of the major challenges in brain network analysis is to further explore the integration of causality and effective connectome analysis in brain network studies. Causality refers to the study of cause-and-effect relationships among variables, which is crucial for understanding the underlying mechanisms of brain function and dysfunction. Effective connectome, on the other hand, represents the causal influences among brain regions, providing a more meaningful characterization of brain organization compared

to traditional functional connectivity based on statistical associations [102]. While we have proposed the DABNet framework to demonstrate the efficacy of modeling causal relationships among ROIs using DAG learning techniques [213], there is still room for improvement. Investigating more advanced causal discovery methods, such as those that can handle potential confounding factors and temporal dependencies, could lead to more accurate and robust estimation of effective brain connectivity. Moreover, extending the framework to incorporate dynamic causal modeling and time-varying effective connectome analysis could provide valuable insights into the temporal evolution of brain networks and their relationship to cognitive processes and clinical outcomes. By leveraging the power of causality and effective connectome analysis, future studies can deepen our understanding of brain organization, identify more reliable biomarkers for neurological and psychiatric disorders, and ultimately advance the field of precision psychiatry.

7.2.2 Few-shot Learning for Extremely Unbalanced Tasks

Another promising direction is to leverage the multi-task learning (MTL) framework proposed in this dissertation for few-shot learning on extremely unbalanced tasks, such as drug abuse prediction. In many real-world scenarios, the number of labeled examples for certain tasks, like drug or alcohol abuse, is often very small compared to the negative examples, making it challenging to train accurate prediction models.

To address this issue, we propose using the MTL framework to learn a shared representation of brain networks across multiple related tasks, capturing important features and patterns. These learned embeddings can then serve as a powerful representation for few-shot learning, where a support set of a few labeled examples is used to train a model to predict on a query set of new, unseen examples. By leveraging the knowledge learned from related tasks through the MTL framework, few-shot learning could potentially improve the performance on these challenging, unbalanced

prediction tasks.

To further enhance the few-shot learning performance, advanced techniques such as meta-learning and prototype-based methods could be explored. Meta-learning algorithms, like Model-Agnostic Meta-Learning (MAML) [64], could be employed to learn an initialization of model parameters that can quickly adapt to new tasks with limited examples. Prototype-based methods, such as Prototypical Networks [171], could be used to learn a metric space where examples from the same class cluster together, enabling effective classification with few examples. Combining these techniques with the MTL framework could lead to powerful few-shot learning models for brain network analysis, opening up new possibilities for studying rare disorders or conditions with limited available data.

7.2.3 Comprehensive and Clinical Evaluation

The last crucial future direction that I want to discuss is to conduct comprehensive evaluations across multiple datasets and tasks while also validating the clinical utility of the proposed methods. This dissertation has demonstrated the superior performance of the developed techniques on several large-scale fMRI datasets, such as PNC [159], ABCD [22], and ABIDE [19], and has provided valuable insights into the complex relationships between brain networks and various cognitive functions and disorders. However, to establish the generalizability and practical value of these methods, it is essential to extend the evaluations to a wider range of neuroimaging datasets, including those from different age groups and clinical populations.

Furthermore, to bridge the gap between research and clinical practice, it is crucial to validate the clinical utility of the proposed deep learning techniques and work towards their deployment in real-world clinical settings. There are three important perspectives to consider in this regard. Firstly, designing more interpretable models is essential to enable clinicians to understand the reasoning behind the model's

predictions and trust its outputs. Secondly, careful verification of the findings from our models with clinicians, neuroscientists, and other domain experts is necessary to ensure the validity and relevance of the insights gained. Thirdly, to facilitate clinical adoption, user-friendly interfaces and visualization tools should be developed to allow clinicians to easily interpret the model predictions and the underlying brain network patterns. By addressing these three key aspects – interpretability, expert validation, and user-friendly tools – we can effectively translate the research findings into clinical practice and ultimately contribute to advancing precision medicine and improving patient care in the field of neuroimaging and brain disorders.

Appendix A

Additional Information for FBNetGen

A.1 1D-CNN Encoder Architecture

We summarize the architecture of 1D-CNN in Table A.1.

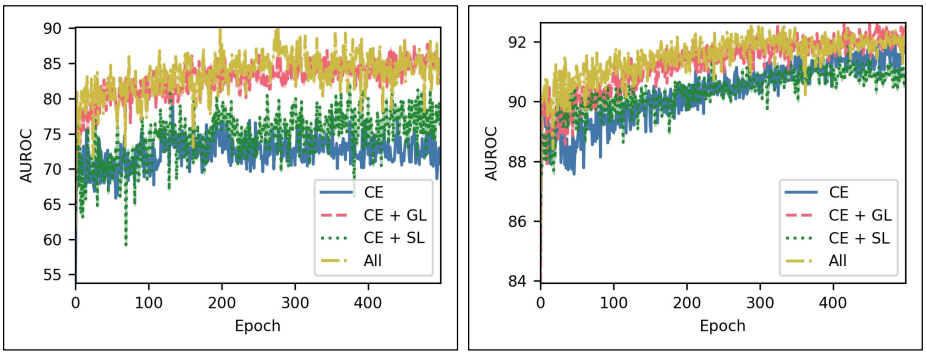
Table A.1: 1D-CNN encoder design.

Layers	Kernal Size	Other Parameters
Conv 1	$1 \times \tau \times 32$	stride=2
Conv 2	$32 \times 8 \times 32$	stride=1
Conv 3	$32 \times 8 \times 16$	stride=1
Max Pool	16	N.A.
Flatten	N.A.	N.A.
Fully Connected 1	N.A.	output=32
ReLU	N.A.	N.A.
Fully Connected 2	N.A.	output=8

A.2 Training Curves of FBNetGen Variants

In Figure A.1, we demonstrate the training curves of different FBNetGen variants. The curves of different variants display similar patterns across datasets. Specifically, it is shown that Group Loss (GL) can achieve pronounced improvement for model's performance, which proves the effectiveness of our loss design in Section 2.3.3. Also,

applying both exterior regularizers (Group Loss and Sparsity Loss) together with the supervised Cross Entropy loss (CE) can consistently achieve the best performance compared with other settings, representing the importance of the mutually restrictive relationships between different regularizers.



(a) Dataset: PNC (b) Dataset: ABCD

Figure A.1: Training curves of FBNetGen variants on two datasets.

A.3 Difference Score T of Functional Modules on Learnable Graph and Pearson Graph

The ranked difference score T, as defined in Eq. (2.16), of functional modules on two kinds of graph, learnable and Pearson, on two datasets are shown in Table A.2 and Table A.3, respectively. Note that the words in bold represent modules that contains more ROIs with significant gender differences according to existing neurobiological findings[160]. The ideal case achieves when the modules with higher difference scores are matched with those known important ones from neuroscience study.

On the PNC dataset, our task-aware learnable graph can obviously highlight the modules with ROIs that are significant different between genders better compared with Pearson graphs. Regarding the ABCD dataset, due to fewer functional modules it contains compared with PNC's, the results of two kinds of graphs are similar. However, our learnable graphs put more emphasize on the functional module

"Somatomotor", which contains ROIs related to auditory functions that are highly differentiated between genders [3].

Overall, the difference score comparison of our learnable graphs and Pearson graphs validates that graphs produced by FBNetGen are task-oriented and can capture more authentic difference between genders than existing famous methods.

Table A.2: Modules' difference score T of our learnable and Pearson graphs on the PNC dataset.

Learnable Graph		**Pearson Graph**	
Module	Difference Score	Module	Difference Score
Memory retrieval	0.083	**Memory retrieval**	0.297
Default mode	0.067	Cingulo-opercular	0.245
Ventral attention	0.064	Subcortical	0.232
Visual	0.054	**Default mode**	0.231
Cingulo-opercular	0.050	**Auditory**	0.206
Fronto-parietal	0.049	Somatomotor Hand	0.181
Subcortical	0.046	Fronto-parietal	0.176
Somatomotor Hand	0.044	Salience	0.164
Cerebellar	0.039	**Ventral attention**	0.155
Somatomotor Mouth	0.036	Visual	0.146
Auditory	0.034	Dorsal attention	0.141
Dorsal attention	0.031	Cerebellar	0.127
Salience	0.030	Somatomotor Mouth	0.114

Table A.3: Modules' difference score T of our learnable and Pearson graphs on the ABCD dataset.

Learnable Graph		**Pearson Graph**	
Module	Difference Score	Module	Difference Score
Default mode	0.301	**Default mode**	0.412
VentralSalience	0.288	**VentralSalience**	0.404
CentralExecutive	0.275	DorsalSalience	0.368
DorsalSalience	0.221	CentralExecutive	0.347
Somatomotor	0.217	Visual	0.322
Visual	0.165	**Somatomotor**	0.301

Appendix B

Additional Information for Brain Network Transformer

B.1 Training Curves of Different Models with or without StratifiedSampling

In Figure B.1, 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.

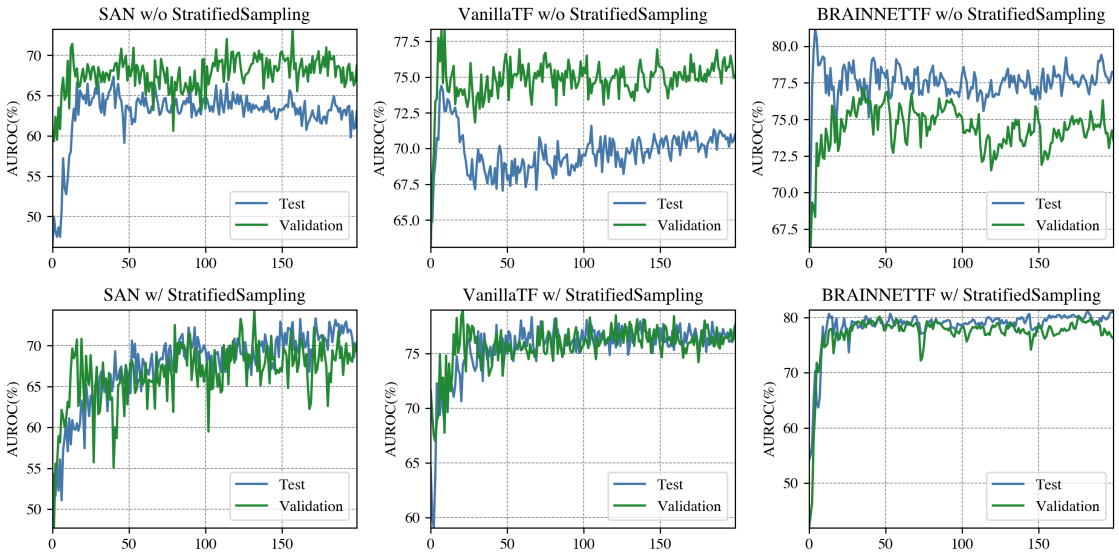


Figure B.1: Training Curves of Different Models with or without StratifiedSampling.

B.2 Transformer Performance with Different Node Features

We compare the performance of Transformer model equipped with different node features. The results are shown in Table B.1, 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 B.1: The Performance (AUROC%) of Transformer with Different Node Features.

B.3 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.

B.4 Proof of Theorem 3.3.1

We state Theorem 3.3.1 here and show the proof details.

Theorem B.4.1. *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*

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

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. (3.2) can be rewritten as:

$$\boldsymbol{P}_1 = \frac{e^{\rho\cos(\theta)}}{e^{\rho\cos(\theta)} + e^{\rho\cos(\theta-\phi)}}, \quad \boldsymbol{P}_2 = \frac{e^{\rho\cos(\theta-\phi)}}{e^{\rho\cos(\theta)} + e^{\rho\cos(\theta-\phi)}}, \tag{B.2}$$

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

is

$$F(\phi) := \frac{1}{V_r}\int_{B_r}\sum_{k=1}^{2}(\boldsymbol{P}_k - \frac{1}{2})^2 d\mathcal{Z} = \frac{1}{\pi r^2}\int_0^r\int_0^{2\pi}\Big(\frac{e^{2\rho\cos(\theta)} + e^{2\rho\cos(\theta-\phi)}}{(e^{\rho\cos(\theta)} + e^{\rho\cos(\theta-\phi)})^2} + \frac{1}{2}\Big)d\theta d\rho. \tag{B.3}$$

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

$$\begin{aligned} f(\frac{\phi}{2} + \frac{\pi}{2} - \theta) &= \frac{e^{2\rho\sin(\frac{\phi}{2}+\theta)} + e^{-2\rho\sin(\frac{\phi}{2}-\theta)}}{(e^{\rho\sin(\frac{\phi}{2}+\theta)} + e^{-\rho\sin(\frac{\phi}{2}-\theta)})^2} \\ &= \frac{e^{2\rho\sin(\frac{\phi}{2}+\theta)} + e^{-2\rho\sin(\frac{\phi}{2}-\theta)}}{e^{2\rho\sin(\frac{\phi}{2}+\theta)} + e^{-2\rho\sin(\frac{\phi}{2}-\theta)} + 2e^{\rho\sin(\frac{\phi}{2}+\theta)-\rho\sin(\frac{\phi}{2}-\theta)}} \\ &= \frac{e^{2\rho\sin(\frac{\phi}{2}-\theta)} + e^{-2\rho\sin(\frac{\phi}{2}+\theta)}}{(e^{\rho\sin(\frac{\phi}{2}-\theta)} + e^{-\rho\sin(\frac{\phi}{2}+\theta)})^2} = f(\frac{\phi}{2} + \frac{\pi}{2} + \theta), \end{aligned} \tag{B.4}$$

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

$$\begin{aligned} \int_{\frac{\phi_1}{2}}^{\frac{\phi_1}{2}+2\pi} f(\rho, \theta, \phi_1)d\theta &= \int_0^{2\pi} f(\rho, \theta, \phi_1)d\theta, \\ \int_{\frac{\phi_1}{2}}^{\frac{\phi_2}{2}+2\pi} f(\rho, \theta, \phi_2)d\theta &= \int_0^{2\pi} f(\rho, \theta, \phi_2)d\theta. \end{aligned} \tag{B.5}$$

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 $\theta = \frac{\phi}{2}$:

$$\begin{aligned} Df(\boldsymbol{v}) &= \frac{\partial f}{\partial \theta} + 2\frac{\partial f}{\partial \phi} \\ &= \frac{2\rho e^{\rho\cos(\theta-\phi)+\rho\cos(\theta)}(e^{\rho\cos(\theta-\phi)} - e^{\rho\cos(\theta)})(\sin(\theta) + \sin(\theta-\phi))}{(e^{\rho\cos(\theta-\phi)} + e^{\rho\cos(\phi)})^3}. \end{aligned} \tag{B.6}$$

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

$$f(\rho, \theta - \frac{\phi_1}{2}, \phi_1) \leq f(\rho, \theta - \frac{\phi_2}{2}, \phi_2) \tag{B.7}$$

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. □

B.5 Proof of Theorem 3.3.2

Theorem 3.3.2 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.

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 [139]:

Definition B.5.1. We consider a regression with r independent main variables:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_r X_r + \epsilon. \tag{B.8}$$

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 $p = 1, 2, ..., r$ $q = 1, 2, ..., s$):

$$\text{SST} = \sum_q (\hat{y}_q - \bar{y})^2, \quad \text{SSR} = \sum_q e_q^2 = \sum_q (\hat{y}_q - \check{y}_q)^2, \quad \text{SSE} = \sum_{q,p} (\hat{x}_{qp} - \bar{x}_p)^2. \tag{B.9}$$

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

defined as:

$$R^2 = \frac{\text{SSE}}{\text{SST}} = 1 - \frac{\text{SSR}}{\text{SST}}. \tag{B.10}$$

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:

$$f_k(\delta) \approx f_k(\delta_T) + \sum_{m=1}^{M} \frac{\partial f_k}{\partial \delta_m}\Big|_{\delta_T} (\delta_m - \delta_{Tm}). \tag{B.11}$$

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

$$\boldsymbol{f}(\delta) \approx \boldsymbol{f}(\delta_T) + \boldsymbol{F}(\delta - \delta_T), \tag{B.12}$$

where $\boldsymbol{F}(\delta - \delta_T)$ stands for the dot product of derivatives and differences of parameters from Eq. (B.11). 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:

$$\check{\boldsymbol{P}} = \boldsymbol{f}(\delta) \approx \boldsymbol{f}(\delta^{(\gamma)}) + \boldsymbol{F}^{(\gamma)}(\delta - \delta^{(\gamma)}), \tag{B.13}$$

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:

$$\boldsymbol{e}(\delta) = \boldsymbol{P} - \boldsymbol{f}(\delta) \approx \boldsymbol{e}(\delta^{(\gamma)}) - \boldsymbol{F}^{(\gamma)}(\delta - \delta^{(\gamma)}). \tag{B.14}$$

Thus, the norm

$$\begin{aligned} S(\delta) &:= \|\boldsymbol{P} - f(\delta)\|^2 = \boldsymbol{e}^\top(\delta)\boldsymbol{e}(\delta) \\ &\approx \boldsymbol{e}^\top(\delta^{(\gamma)})\boldsymbol{e}(\delta^{(\gamma)}) - 2\boldsymbol{e}^\top(\delta^{(\gamma)})\boldsymbol{F}^{(\gamma)}(\delta - \delta^{(\gamma)}) + (\delta - \delta^{(\gamma)})^\top \boldsymbol{F}^{(\gamma)\top}\boldsymbol{F}^{(\gamma)}(\delta - \delta^{(\gamma)}). \end{aligned} \tag{B.15}$$

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

$$\delta - \delta^{(\gamma)} = (\boldsymbol{F}^{(\gamma)\top}\boldsymbol{F}^{(\gamma)})^{-1}\boldsymbol{F}^{(\gamma)\top}\boldsymbol{e}(\delta^{(\gamma)}) = \zeta^{(\gamma)}. \tag{B.16}$$

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

$$\delta^{(\gamma+1)} = \delta^{(\gamma)} + \zeta^{(\gamma)}. \tag{B.17}$$

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., [25, 38]. We do not discuss this sophisticated method as it is out of the scope of this paper.

Comparing R^2 by Variance Inflation Factor

The proof of Theorem 3.3.2 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 B.5.4), and consequently (b) higher regression accuracy increases *appraisal accuracy*, the accuracy when appraising an estimated value in *hypothesis testing* (Theorem B.5.7).

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* [139]:

$$\mathrm{E}(\check{\boldsymbol{P}}) = \boldsymbol{P}_T = \mathrm{E}(\boldsymbol{P}). \tag{B.18}$$

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 B.5.2. The variance inflation factor $(\mathrm{VIF})_p$ is defined as:

$$(\mathrm{VIF})_p = \frac{1}{(1 - R_p^2)}, \tag{B.19}$$

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. (B.8).

Remark B.5.3. We discuss more details about VIF in the following context [139]. 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. (B.8), the estimation $\check{\beta}_p$ of regression coefficients β_p are

obtained by Ordinary Least Squares (OLS):

$$\check{\beta} = (\hat{X}^\top \hat{X})^{-1} \hat{X}^\top \hat{y}. \tag{B.20}$$

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

$$\check{y}_q^* = \frac{\check{y}_q - \bar{y}}{\sigma_y}, \quad \hat{x}_{qp}^* = \sigma_q^{-1}(\hat{x}_{pq} - \bar{x}_p), \tag{B.21}$$

and

$$\check{\beta}_q^* = \check{\beta}_q \frac{\sigma_q}{\sigma_y}, \quad \check{y}^* = \check{\beta}_0^* + \check{\beta}_1^* X_1^* + \check{\beta}_2^* X_2^* + \cdots + \check{\beta}_r^* X_r^*. \tag{B.22}$$

Similarly to Eq. (B.20), standardized estimation of regression coefficients are equal to

$$\check{\beta}^* = (\check{X}^{*\top} \check{X}^*)^{-1} \check{X}^{*\top} \check{y}^*. \tag{B.23}$$

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

$$\sigma_{\check{\beta}}^2 = \sigma^2 (X^\top X)^{-1}, \quad \sigma^2 = \sum_{q=1}^{s} (\check{y}_q - \bar{y})^2, \tag{B.24}$$

where σ^2 is the *error term variance* for X (cf. Definition B.5.1). After standardization, it is noted that $X^{*\top} X^*$ is just the correlation matrix r_{XX} of X^*. Hence, by Eq. (B.24) we obtain:

$$\sigma_{\check{\beta}^*}^2 = (\sigma^*)^2 r_{XX}^{-1}. \tag{B.25}$$

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

$$\sigma^2_{\check{\beta}_p^*} = (\sigma^*)^2(\text{VIF})_p. \tag{B.26}$$

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 [139]

$$\sigma^2_{\check{\beta}_p^*} = \frac{1}{1-R_p^2}\Big[\frac{\sigma^2}{\sum_q (x_{qp}-\bar{x}_p)^2}\Big]. \tag{B.27}$$

With the previous discussion, we conclude that

$$(\text{VIF})_p = \frac{1}{(1-R_p^2)}, \tag{B.28}$$

where R_p^2 is defined in B.5.2.

Theorem B.5.4. *Let*

$$\mathit{VIF} = \frac{\sum_{p=1}^{r}(\mathit{VIF})_p}{r-1}, \tag{B.29}$$

where r denotes the number of variables in Eq. (B.8)*. 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,

$$R^2 = \frac{\text{SSE}}{\text{SST}} = \frac{\sum_{q=1}^{s}(\check{y}_q-\bar{y})^2}{\sum_{q=1}^{s}(y_q-\bar{y})^2} = \sum_{q=1}^{s}(\check{y}_q^*)^2. \tag{B.30}$$

Substituting Eq. (B.22) into the above identity, we have

$$\sum_{q=1}^{s}(\check{y}_q^*)^2 = \sum_{q=1}^{s}(\check{X}_q^*\check{\beta}^*)^2 = (X_q^*\check{\beta}^*)^\top X_q^*\check{\beta}^*, \tag{B.31}$$

and by Eq. (B.23), we conclude that

$$R^2 = (r_{XY})^\top (r_{XX})^{-1} r_{XY}. \tag{B.32}$$

As the finial step, we compute R_p^2 from Definition B.5.2 by Eq. (B.32). It should be noted that according to Definition B.5.2, 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 B.5.5. 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 B.5.1 (see [139, 1] for more details).

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 B.5. We first introduce *mean squared error* (MSE) commonly used in statistics [53]:

Definition B.5.6. Recall that the residue $e_q = (\hat{y}_q - \check{y}_q)$ from Definition B.5.1. Then,

$$\text{MSE} = \frac{1}{s}\sum_{q=1}^{s}(\hat{y}_q - \check{y}_q)^2 = \frac{1}{s}\sum_{q=1}^{s}(e_q)^2 = \frac{1}{s}\boldsymbol{e}^\top \boldsymbol{e}. \tag{B.33}$$

As mentioned in B.5, a small coefficient of determination R^2 indicates a large SSR and hence leads to a large MSE. As a result of Theorem B.5.4, 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 B.5.4) guarantees a lower MSE and therefore increases the appraisal accuracy.

Theorem B.5.7. *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

$$H_0 : \check{\boldsymbol{P}} = \boldsymbol{P}_T, \quad H_1 : \check{\boldsymbol{P}} \neq \boldsymbol{P}_T, \tag{B.34}$$

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 [53]. 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:

$$g(\theta) = P_\theta(\check{\boldsymbol{P}} \in W) = \int_W f(T_{\boldsymbol{E}_k}(\boldsymbol{Z}_i))d\boldsymbol{Z}_i \leq \alpha_{\boldsymbol{E}_k}, \quad \theta = H_0 \cup H_1, \tag{B.35}$$

where the significance level $\alpha_{\boldsymbol{E}_k}$ is the upper bound of the probability of making mistakes (formally called *type I error*) [53].

By Theorem B.5.4 and Remark B.5.6, 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 $\boldsymbol{E}'_k$:

$$\int_W f(T_{\boldsymbol{E}_k}(\boldsymbol{Z}_i))d\boldsymbol{Z}_i \leq \int_W f(T_{\boldsymbol{E}'_k}(\boldsymbol{Z}_i))d\boldsymbol{Z}_i. \tag{B.36}$$

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. □

B.6 Running Time

Table B.2 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 [63] 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 B.2: 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**

B.7 Number of Parameters

Table B.3: 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	57.76M	186.7M
Graphormer	1.23M	1.66M
VanillaTF	15.6M	32.7M
BrainNetTF	4.0M	11.2M

B.8 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

Table B.4: 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

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.

B.9 Software Version

B.10 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

$$d = \sum_{i}^{K} \sum_{j}^{V} \frac{|P_{ij}^{female} - P_{ij}^{male}|}{KV}, \tag{B.37}$$

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 B.5: 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

Appendix C

Additional Information for R-Mixup

C.1 Covariance, Correlation and Positive Definite Matrices

We provide detailed definitions on covariance, correlation and positive definite matrices with necessary properties here.

Definition C.1.1. Let $X = (X_i) = (x_{ik})$ with $i = 1, ..., n$ and $k = 1, ..., t$ be t-dimensional vectors of n variables. The corresponding *covariance matrix* $\mathrm{Cov}(X)$ is defined as

$$\begin{aligned} \mathrm{Cov}(X)_{ij} &= \frac{1}{t}\Big(\sum_k (x_{ik} - E(X_i))(x_{jk} - E(X_j))\Big) \\ &= E(X_i X_j) - E(X_i)E(X_j). \end{aligned} \tag{C.1}$$

The *correlation matrix* is normalized as:

$$\begin{aligned}\mathrm{Cor}(X) =&\mathrm{diag}(\frac{1}{\sqrt{\mathrm{Cov}(X)_{11}}},...,\frac{1}{\sqrt{\mathrm{Cov}(X)_{vv}}})\cdot \\ &\mathrm{Cov}(X)\cdot\mathrm{diag}(\frac{1}{\sqrt{\mathrm{Cov}(X)_{11}}},...,\frac{1}{\sqrt{\mathrm{Cov}(X)_{vv}}}).\end{aligned} \tag{C.2}$$

Expressed by matrix entries, we restore the familiar *Pearson correlation coefficients*:

$$\mathrm{Cor}(X)_{ij} = \frac{\mathrm{Cov}(X)_{ij}}{\sqrt{\mathrm{Cov}(X)_{ii}}\sqrt{\mathrm{Cov}(X)_{jj}}}. \tag{C.3}$$

Remark C.1.2. It should be noted that to make the definition of $\mathrm{Cor}(X)$ valid, $\mathrm{Cov}(X)_{ii} \neq 0$ for all i. Since

$$\mathrm{Cov}(X)_{ii} = E(X_iX_i) - E(X_i)E(X_i) = \frac{1}{t}\sum_k x_{ik}^2 - \Big(\frac{1}{t}\sum_k x_{ik}\Big)^2, \tag{C.4}$$

the geometric mean inequality says that $\mathrm{Cov}(X)_{ii}$ vanishes only when x_{ik} are identical, which does not happen in our case.

Definition C.1.3. A symmetric $n \times n$ matrix S is *positive semi-definite* if for any vector $u \in \mathbb{R}^n$, $u^TSu \geq 0$. Equivalently, this means that the eigenvalues of S are all nonnegative. If the inequality holds strictly, S is said to be *positive definite*, or *symmetric positive definite*, or SPD for short.

Proposition C.1.4. *Covariance and correlation matrices are positive semi-definite. A necessary condition for them to be positive definite is that the length of each sample is no less than the number of variables, i.e., $t \geq n$.*

Proof. Recall Eq.(C.1) from Definition C.1.1, let us consider column vectors $Y_k = (x_{ik} - E(X_i))$. Then $\mathrm{Cov}(X) = \frac{1}{t}\sum_k Y_kY_k^T$. Given any vector $u \in \mathbb{R}^n$,

$$u^T\mathrm{Cov}(X)u = \frac{1}{t}\sum_i u^TY_kY_k^Tu = \frac{1}{t}\sum_i (Y_k^Tu)^2 \geq 0. \tag{C.5}$$

On the other hand, by Eq.(C.2)

$$\begin{aligned} u^T \mathrm{Cor}(X) u &= u^T \mathrm{diag}(\frac{1}{\sqrt{\mathrm{Cov}(X)_{11}}}, ..., \frac{1}{\sqrt{\mathrm{Cov}(X)_{vv}}}) \cdot \\ &\quad \mathrm{Cov}(X) \cdot \mathrm{diag}(\frac{1}{\sqrt{\mathrm{Cov}(X)_{11}}}, ..., \frac{1}{\sqrt{\mathrm{Cov}(X)_{vv}}}) u \\ &= \tilde{u}^T \mathrm{Cov}(X) \tilde{u} \geq 0. \end{aligned} \tag{C.6}$$

If $\{Y_k\}_{k=1}^t$ spans the whole vector space $\mathbb{R}^n$, in which case t must be no less than n, then $\mathrm{Cov}(X)$ is positive definite. Otherwise, there must be some vector u perpendicular to all Y_k, which leads to $\sum_i (Y_k^T u)^2 = 0$. □

C.2 Geodesics and Swelling Effect

We list the geodesic equation and Riemannian distance function induced from log-Euclidean metric on $\mathrm{Sym}^+(n)$ here followed by a rigorous proof on the swelling effect.

Definition C.2.1. Let $\mathrm{Sym}^+(n)$ denote the manifold of positive definite matrices equipped with the *log-Euclidean metric*. Analytically, the induced distance function reads

$$d(S_i, S_j) = \|\log S_i - \log S_j\|, \tag{C.7}$$

which measures the distance between different two points $S_i, S_j \in \mathrm{Sym}^+(n)$, with the following geodesic connecting two points:

$$\gamma(\lambda) = \exp((1-\lambda) \log S_i + \lambda \log S_j). \tag{C.8}$$

Detailed derivation of the geodesics equation can be found in [158, 124, 72].

Proposition C.2.2 (Swelling Effect)**.** *Given arbitrary $S_i, S_j \in Sym^+(n)$, then*

$$\begin{aligned} &\det\left(\exp((1-\lambda)\log S_i + \lambda \log S_j)\right) \\ \leq &\det\left(((1-\lambda)\log S_i + \lambda \log S_j)\right). \end{aligned} \tag{C.9}$$

Especially,

$$\begin{aligned} &\min\{\det S_i, \det S_j\} \\ \leq &\det\left(\exp((1-\lambda)\log S_i + \lambda \log S_j)\right) \leq \max\{\det S_i, \det S_j\}, \end{aligned} \tag{C.10}$$

while $\det\left((1-\lambda)\log S_i + \lambda \log S_j\right)$ *would exceed the determinants of both* S_i *and* S_j *as shown in Figure 5.2 in the main text.*

Proof. To prove the first inequality, we note one basic fact of matrix exponential: $\det(\exp(A)) = \exp(\mathrm{Tr}A)$. Thus,

$$\begin{aligned} &\det\left(\exp((1-\lambda)\log S_i + \lambda \log S_j)\right) \\ =&\exp\left(\mathrm{Tr}((1-\lambda)\log S_i + \lambda \log S_j)\right) \\ =&\exp\left((1-\lambda)\mathrm{Tr}\log S_i\right)\exp\left(\lambda \mathrm{Tr}\log S_j\right)) \\ =&(\exp \mathrm{Tr}\log S_i)^{1-\lambda}(\exp \mathrm{Tr}\log S_j)^{\lambda} \\ =&(\det S_i)^{1-\lambda}(\det S_2)^{\lambda}. \end{aligned} \tag{C.11}$$

Then we make use of the following identity of n-dimensional Gaussian integral:

$$\int \exp(-\frac{1}{2}\boldsymbol{x}^T S\boldsymbol{x})d\boldsymbol{x} = \sqrt{\frac{(2\pi)^n}{\det S}}, \tag{C.12}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and $S \in \mathrm{Sym}^+(n)$. In our case,

$$\begin{aligned}
&\sqrt{\frac{(2\pi)^n}{\det\left(((1-\lambda)S_i + \lambda S_j)\right)}} \\
=&\int \exp(-\frac{1}{2}\boldsymbol{x}^T((1-\lambda)S_i + \lambda S_j)\boldsymbol{x})d\boldsymbol{x} \\
=&\int \Big(\exp(-\frac{1}{2}\boldsymbol{x}^T S_i\boldsymbol{x})\Big)^{1-\lambda}\Big(\exp(-\frac{1}{2}\boldsymbol{x}^T S_j\boldsymbol{x})\Big)^{\lambda} d\boldsymbol{x} \\
\leq&\Big(\int \exp(-\frac{1}{2}\boldsymbol{x}^T S_i\boldsymbol{x})d\boldsymbol{x}\Big)^{1-\lambda}\Big(\int \exp(-\frac{1}{2}\boldsymbol{x}^T S_j\boldsymbol{x})d\boldsymbol{x}\Big)^{\lambda} \\
=&\sqrt{\frac{(2\pi)^n}{(\det S_i)^{1-\lambda}(\det S_j)^{\lambda}}}.
\end{aligned} \tag{C.13}$$

We use Hölder's inequality in the last step from above, which yields

$$\begin{aligned}
&\det\left(\exp((1-\lambda)\log S_i + \lambda \log S_j)\right) \\
=&(\det S_i)^{1-\lambda}(\det S_j)^{\lambda} \leq \det\left(((1-\lambda)S_i + \lambda S_j)\right).
\end{aligned} \tag{C.14}$$

To prove the second inequality, let us assume $\det S_i \leq \det S_j$ and let $a = \det S_j / \det S_i \geq 1$. It is straightforward to check that $a^\lambda - 1 \leq (a-1)\lambda$ when $0 \leq \lambda \leq 1$. This fact indicates that

$$\begin{aligned}
&(\frac{\det S_j}{\det S_i})^{\lambda} - 1 \leq (\frac{\det S_j}{\det S_i} - 1)\lambda \\
\Longrightarrow\ &(\frac{\det S_j}{\det S_i})^{\lambda} \leq (\frac{\det S_j}{\det S_i})\lambda + (1-\lambda) \\
\Longrightarrow\ &(\det S_i)^{1-\lambda}(\det S_j)^{\lambda} \leq (\det S_i)(1-\lambda) + (\det S_j)\lambda,
\end{aligned} \tag{C.15}$$

and finishes the proof. □

C.3 Kernel Regression on $\mathbf{Sym}^+(n)$

We now present the proof details of Theorem 5.3.3 from the main text. To begin with, we introduce a method from heat kernel theory [12] to generalize to Euclidean

Gauss kernel

$$K_E(S_i, \tilde{S}) = \frac{1}{(2\pi\sigma^2)^{n^2/2}} \exp(-\frac{1}{2\sigma^2}\|S_i - \hat{S}\|^2) \tag{C.16}$$

on $\mathrm{Sym}^+(n)$ over which our samples are distributed. The notion of *geodesic regression* [16, 66] would also become apparent as we move forward. Let us first consider the following classical heat equation on Euclidean space

$$\Big(\frac{\partial}{\partial t} - \sum_i \frac{\partial^2}{\partial x_i^2}\Big) f = \frac{\partial f}{\partial t} + \Delta f = 0, \tag{C.17}$$

where $\Delta = \sum_i \frac{\partial^2}{\partial x_i^2}$ is the Laplacian. A solution $f(x,t)$ to this equation is interpreted as the temperature at position x and time t. Substituting $t = \sigma^2/2$ into Eq.(5.9), it can be check by definition that the function

$$K_t(x, y) = \frac{1}{(4\pi t)^{n^2/2}} \exp(-\frac{1}{4t}\|x - y\|^2) \tag{C.18}$$

solves Eq.(C.17). It is called the *fundamental solution* or *heat kernel* as any other solutions to Eq.(C.17) can be written as the convolution with a certain function $f(y)$:

$$f(x,t) = \int_{\mathbb{R}^{n\times n}} K_t(x,y) f(y) dy. \tag{C.19}$$

Based on this fact, it is natural to define Riemannian Gauss kernel as the fundamental solution to heat equation on Riemannian manifolds. To this end, we need to replace the Laplacian on Euclidean spaces by *Laplace–Beltrami operator*, still denoted by Δ, on manifolds. The formal definition of this operator is unnecessary here and we recommend interested readers to [12, 72] for more details. It is enough to known its local coordinate expression for our purpose. Specifically, being different from Euclidean spaces with a standard and explicit *coordinate system*, i.e., any vector $\boldsymbol{x} \in \mathbb{R}^n$

can be explicitly expressed by its components (coordinates) x_i, the coordinates of points p in a manifold M always need being defined exclusively depending on the concerned manifold. In the most general case, it is only known that manifolds admit local coordinate parametrizations for its local regions as they resemble Euclidean spaces. Expressed by any local coordinates,

$$\Delta f = \sum_{i,j} g^{ij}\Big(\frac{\partial^2 f}{\partial x_i^2} - \Gamma_{ij}^k \frac{\partial f}{\partial x_k}\Big) \tag{C.20}$$

where g^{ij}, Γ_{ij}^k are called dual metric and Christoffel symbols and are all determined by the Riemannian metric [158, 124]. Now we wish to analyze the heat equation expressed by local coordinates. However due to its intricate form involving the Riemannian metric, it is generally impossible to solve the equation analytically. Even though, the following theorem is established in heat kernel theory using advanced tools from differential geometry:

Theorem C.3.1. *[12] Let M be a complete Riemannian manifold, then there exists a function $K_t(p,q)$, called heat kernel, with the following properties*

1. *$K_t(p,q) = K_t(q,p)$ for all $p, q \in M$.*
2. *$\lim_{t\to 0} K_t(p,q)$ equals the Dirac delta function $\delta_x(y)$.*
3. *$K_t(p,q)$ is positive definite and solves the heat equation.*
4. *$K_t(p,q) = \int_M K_{t-s}(p,p')K_s(p',q)dp'$ for any $s > 0$.*

We are only interested in the third property as its confirms that the heat kernel truly determines a feature map from $\mathrm{Sym}^+(n)$ into a higher dimensional feature space [164]. Let $\boldsymbol{y} = (y_i)$ denote the column vector consisting of training data labels, let $\boldsymbol{K}_{\tilde{S}}$ denote the column vector consisting of $K_R(S_i, \tilde{S})$ and let $G = (K_R(S_i, S_j)$ denote the kernel matrix evaluated by K_R on the data set. Then the predictor function regressed

through *kernel ridge regression* is

$$\tilde{m}(\tilde{S}) = \boldsymbol{y}^T G^{-1} \boldsymbol{K}_{\tilde{S}}. \tag{C.21}$$

As a reminder, if $\det G = 0$ (when does not happen here since the kernel function is positive definite), a regularization $\zeta \geq 0$ can be chosen as a trade-off between weights and square errors when optimizing the regression. The log-Euclidean metric is now explicitly used in evaluating $\boldsymbol{K}_{\tilde{S}}$ as well as G. On the other hand, a vanilla geodesic regression could be intuitively treat as the multi-linear regression on manifold with the Riemannian metric substituting for the Euclidean metric, which is fairly easy to deal with when we have a coordinate system and this is what we are going to do in the following proof.

Theorem C.3.2. *For $Sym^+(n)$ with log-Euclidean metric and estimators $\tilde{m}$ obtained with either geodesic regression or Gaussian kernel regression on samples. Augmented data from Riemannian geodesics bear the mean square error no more than those from straight lines.*

Proof. We first note the fact that the exponential and logarithm functions

$$\exp : \mathrm{Sym}(n) \to \mathrm{Sym}^+(n), \quad \log : \mathrm{Sym}^+(n) \to \mathrm{Sym}(n) \tag{C.22}$$

are *isometries* between $\mathrm{Sym}^+(n)$ and $\mathrm{Sym}(n)$. That is: (a) they are bijective and (b) preserve the Riemannian distance functions:

$$\|H_i - H_j\| = d(\exp H_i, \exp H_j), \quad d(S_i, S_i) = \|\log S_i - \log S_j\| \tag{C.23}$$

for any $S_i, S_j \in \mathrm{Sym}^+(n)$ and any $H_i, H_j \in \mathrm{Sym}(n)$. A detailed proof on the RHS equation from can be found in [72] and with the bijectivity of exp and log, we obtain the LHS equation from above. Besides, being defined as collection of all symmetric

$n \times n$ matrices, $\mathrm{Sym}(n)$ is an $\frac{1}{2}n(n-1)$-dimensional Euclidean space with a standard coordinate system introduced above. Combining with the logarithm function $\log : \mathrm{Sym}^+(n) \to \mathrm{Sym}(n)$, then we obtain a coordinate system for $\mathrm{Sym}^+(n)$ within which we can express the heat equation Eq.(C.17) explicitly.

Since log is an isometry and since $\mathrm{Sym}(n)$ is a Euclidean space, as a basic result in Riemannian geometry, the Christoffel symbols Γ_{ij}^k in Eq.(C.20) vanishes [124] in the defined coordinate system and hence the Laplace–Beltrami operator degenerates to the common Laplacian. As a result, the fundamental solution is exactly Eq.(C.18) expressed by the coordinates. After taking the inverse map exp, the Euclidean distance is replaced by Riemannian distance in Eq.(C.18). Hence,

$$K_R(S_i, \hat{S}) = \frac{1}{(2\pi\sigma^2)^{\frac{n(n-1)}{4}}} \exp\left(-\frac{1}{2\sigma^2} d(S_i, \hat{S})^2\right), \tag{C.24}$$

is the heat kernel on $\mathrm{Sym}^+(n)$ with the property being positive definite by Theorem C.3.1.

Recall that the Riemannian geodesic of log-Euclidean metric is

$$\gamma(\lambda) = \tilde{S} = \exp\left((1-\lambda)\log S_i + \lambda \log S_j\right). \tag{C.25}$$

Its coordinate representations is then

$$\log(\gamma(\lambda)) = (1-\lambda)\log S_i + \lambda \log S_j, \tag{C.26}$$

which is a straight line connecting $\log S_i$ and $\log S_j \in \mathrm{Sym}(n)$. As a contrary, the coordinate representation of

$$\eta(\lambda) = \tilde{S}' = (1-\lambda)S_i + \lambda S_j \tag{C.27}$$

is highly curved as

$$\log(\eta(\lambda)) = (\log(1-\lambda)S_i + \lambda S_j)\,. \tag{C.28}$$

Since Sym(n) is an Euclidean space and since we verified above that the function log is an isometry, conducting geodesic regression for the samples $\{(S_i, y_i)|i = 1, ..., N\}$ is merely solving the linear model of $\{(\log S_i, y_i)|i = 1, ..., N\}$. Since the total sample number N in our case is less than the dimension of the ambient Euclidean space n^2, the optimal solution is just a hyperplane encompassing all samples as well as those synthesized via Eq.(C.26). However, curves like Eq.(C.28) are manifestly deviated from the regression hyperplane which leads to large square loss.

To verify the case involving Gaussian kernel, we make use of the following operator inequalities [21]:

$$\log((1-\lambda)S_i + \lambda S_j) \geq (1-\lambda)\log S_i + \lambda \log S_j. \tag{C.29}$$

Intuitively, the logarithm is a concave function on $(0, +\infty)$, which is generalized to hold in the setting of positive semidefinite matrices with $A \geq B$ meaning $A - B$ is positive semidefinite. For simplicity, we only analyze Eq.(C.21) for a pair of samples S_i, S_j as an augmented sample $\tilde{S}$ is coined in this way through our mixup method. In statistics, penalty functions [206] can be employed weaken the influence of other samples and achieve this effect. With these preparation,

$$\begin{aligned}\tilde{m}(S) = \boldsymbol{y}^T G^{-1} \boldsymbol{K}_S &= (y_i, y_j) \begin{pmatrix} K_R(S_i, S_i) & K_R(S_i, S_j) \\ K_R(S_j, S_i) & K_R(S_i, S_i) \end{pmatrix}^{-1} \begin{pmatrix} K_R(S_i, S) \\ K_R(S_j, S) \end{pmatrix} \\ &= \frac{1}{1 - K_{ij}^2}\Big((y_i - K_{ij}y_j)K_{i,S} + (y_j - K_{ij}y_i)K_{j,S})\Big),\end{aligned} \tag{C.30}$$

where $K_{ij} = \exp\left(-\frac{1}{2\sigma^2}d(S_i, \hat{S})^2\right)$ is an abbreviation for the non-normalized Gaussian

distribution of log-Euclidean distance with $K_{i,S}$ being denoted analogously. Substituting $\tilde{S}$ and $\tilde{S}'$ from Eq.(C.25) and Eq.(C.27) into the above equation, we then compare the estimators with $\tilde{y} = (1-\lambda)y_i + \lambda y_j$ directly.

For predictions of $\tilde{S}$, we note that

$$K_{ij} = \exp\left(-\frac{1}{2\sigma^2}\|\log S_i - \log S_j\|\right), \tag{C.31}$$

$$K_{i,\tilde{S}} = \exp\left(-\frac{1}{2\sigma^2}\lambda\|\log S_i - \log S_j\|\right) = K_{ij}^{\lambda} \tag{C.32}$$

$$K_{j,\tilde{S}} = \exp\left(-\frac{1}{2\sigma^2}(1-\lambda)\|\log S_i - \log S_j\|\right) = K_{ij}^{1-\lambda} \tag{C.33}$$

with

$$\tilde{m}(\tilde{S}) = \frac{1}{1-K_{ij}^2}\Big(K_{ij}^{\lambda}(y_i - K_{ij}y_j) + K_{ij}^{1-\lambda}(y_j - K_{ij}y_i))\Big) \tag{C.34}$$

being a *concave function* for $\lambda \in [0,1]$. This can be demonstrated by examining that the second order derivative

$$\frac{d^2\tilde{m}(\tilde{S}(\lambda))}{d\lambda^2} = \frac{\ln^2 K_{ij}}{K_{ij}^{\lambda}(1-K_{ij}^2)}\Big((K_{ij} - K_{ij}^{2\lambda+1})y_j + (K_{ij}^{2\lambda} - K_{ij}^2)y_i\Big), \tag{C.35}$$

which is nonnegative because $K_{ij} \leq 1$ and $K_{ij} - K_{ij}^{2\lambda+1}, K_{ij}^{2\lambda} - K_{ij}^2 \geq 0$. As a result, $\tilde{m}(\tilde{S}) \leq \tilde{y}$. On the other hand,

$$K_{i,\tilde{S}'} = \exp\left(-\frac{1}{2\sigma^2}\|\log((1-\lambda)S_i - \lambda S_j) - \log S_i\|\right) \tag{C.36}$$

$$K_{j,\tilde{S}'} = \exp\left(-\frac{1}{2\sigma^2}\|\log((1-\lambda)S_i - \lambda S_j) - \log S_j\|\right) \tag{C.37}$$

are intricate as the linear combination of matrices $((1-\lambda)S_i - \lambda S_j)$ does not commute with the logarithm. Despite of this difficulty, we are still above to compare $\tilde{m}(\tilde{S})$ and

$\tilde{m}(\tilde{S}')$ based on their general expansion in Eq.(C.30). By (C.29),

$$\begin{aligned} &\log((1-\lambda)S_i + \lambda S_j) - \log S_i \geq \lambda(\log S_i + \log S_j) \\ \implies &\|\log((1-\lambda)S_i + \lambda S_j) - \log S_i\| \geq \|\lambda(\log S_i + \log S_j)\| \\ \implies &K_{i,\tilde{S}'} = \exp\left(-\frac{1}{2\sigma^2}\|\log((1-\lambda)S_i - \lambda S_j) - \log S_i\|\right) \\ &\leq \exp\left(-\frac{1}{2\sigma^2}\lambda\|\log S_i - \log S_j\|\right) = K_{i,\tilde{S}}. \end{aligned} \tag{C.38}$$

The second inequality is due to the fact that the operator norm $\|$ - $\|$ equals the largest eigenvalue of any positive semidefinite operator. Similar argument also implies case with S_j. Together with the concavity of $\tilde{m}(\tilde{S})$, Eq.(C.30) and the range of our labels, we conclude that

$$0 \leq \tilde{m}(\tilde{S}') \leq \tilde{m}(\tilde{S}) \leq \tilde{y} \implies \sum(\tilde{m}(\tilde{S}) - \tilde{y})^2 \leq (\tilde{m}(\tilde{S}') - \tilde{y})^2, \tag{C.39}$$

which are finally summed over the samples to show that the square error of estimation using geodesics is no more than that using straight lines on $\mathrm{Sym}^+(n)$. □

Remark C.3.3. For affine-invariant metric, it has been shown that the induced *Riemannian curvature tensor* R is nonzero [158, 176] and hence it is impossible to find coordinate systems within which $\Gamma_{ij}^k = 0$ [124]. Therefore, the fundamental solution to the heat equation can never take in the concise form as Eq.(C.24) and Theorem 5.3.3 becomes invalid to appraise the case when using affine-invariant metric.

C.4 Running Time on three smaller datasets

As shown in Figure C.1, on the smaller datasets, PNC, ABIDE, and TCGA-Cancer, there is no significant difference in elapsed time between the different methods. Notably, the proposed R-Mixup is magically faster than C-Mixup on the TCGA-Cancer

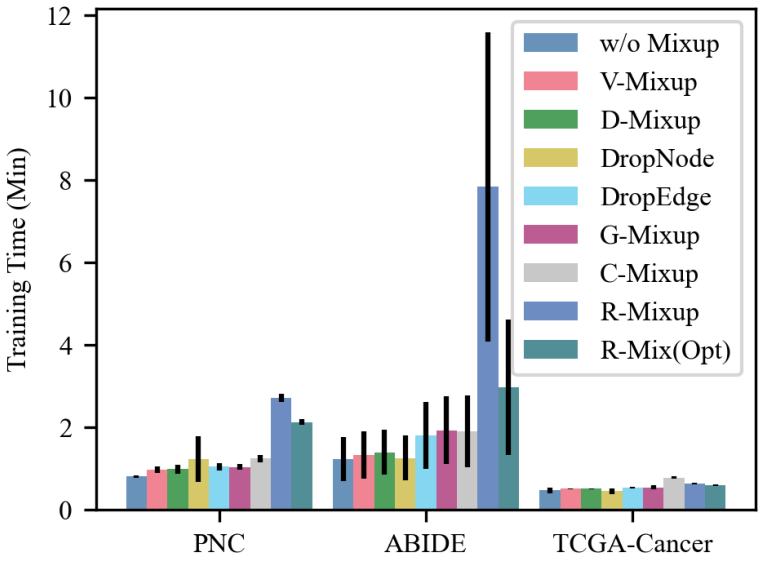


Figure C.1: Running Time in PNC, ABIDE and TCGA-Cancer. R-Mixup is the original version of our method while R-Mix(Opt) is the proposed optimized version in Section 5.3.5.

dataset. This is mainly due to the small node size of TCGA-Cancer, which reduces the main barrier of the eigenvalue decomposition in R-Mixup, while the time cost of the sampling operation in the C-Mixup baseline does not change dynamically with the node size.

C.5 Code Implementation

```
import torch
import numpy as np

wdef tensor_log(t):
    # condition: t is symmetric.
    s, u = torch.linalg.eigh(t)
    s[s <= 0] = 1e-8
    return u @ torch.diag_embed(torch.log(s)) @ u.permute(0, 2, 1)

```

```python
def tensor_exp(t):
    # condition: t is symmetric.
    s, u = torch.linalg.eigh(t)
    return u @ torch.diag_embed(torch.exp(s)) @ u.permute(0, 2, 1)

def r_mixup(x, y, alpha=1.0, device='cuda'):
    if alpha > 0:
        lam = np.random.beta(alpha, alpha)
    else:
        lam = 1
    batch_size = y.size()[0]
    index = torch.randperm(batch_size).to(device)
    x = tensor_log(x)
    x = lam * x + (1 - lam) * x[index, :]
    y = lam * y + (1 - lam) * y[index]
    return tensor_exp(x), y
```

Listing C.1: Python Example

C.6 GCN Backbone Performance

The performance of models with the GCN backbones can be found in Table C.1.

Table C.1: Overall performance comparison based on the GCN backbone. The best results are in bold, and the second best results are underlined. The ↑ indicates a higher metric value is better and ↓ indicates a lower one is better.

Method	ABCD-BioGender		ABCD-Cog	PNC		ABIDE		TCGA-Cancer	
	AUROC↑	Accuracy↑	MSE↓	AUROC↑	Accuracy↑	AUROC↑	Accuracy↑	Precision↑	Recall↑
w/o Mixup	78.82±0.62	71.55±0.43	80.85±4.69	59.14±5.66	60.00±4.72	55.09±6.91	55.20±6.18	30.49±6.89	40.83±6.18
V-Mixup	81.47±0.79	73.97±0.79	80.35±3.09	63.63±3.80	61.76±3.25	58.49±6.58	56.40±3.91	38.50±9.22	46.67±11.18
D-Mixup	81.30±0.35	73.67±0.39	80.90±9.48	58.68±6.24	58.43±5.99	60.19±6.58	55.40±6.15	37.67±5.47	**50.00±4.17**
DropNode	80.77±2.02	73.18±2.09	88.11±10.59	63.65±5.04	61.57±5.16	59.49±4.99	56.60±5.55	29.58±7.69	39.17±10.46
DropEdge	79.98±1.54	72.23±1.37	85.98±2.31	56.61±2.72	56.67±2.89	56.58±6.78	54.80±4.76	39.44±7.72	**50.00±7.80**
G-Mixup	81.30±1.07	73.90±0.86	81.28±3.46	57.25±3.75	57.45±2.91	62.43±2.94	**60.40±3.44**	38.64±8.47	49.17±9.03
C-Mixup	81.62±1.65	73.62±1.80	78.86±3.51	60.88±7.24	58.24±7.61	60.22±9.32	57.40±5.32	34.17±11.74	46.67±15.14
R-Mixup	**82.85±1.86**	**75.86±1.88**	**74.88±2.03**	**64.39±5.05**	**62.31±3.32**	**63.03±5.58**	59.67±5.96	**44.78±8.64**	48.44±8.61

C.7 Case Study About Arbitrarily Incorrect Label Problem

To verify how R-Mixup migrate the *arbitrarily incorrect label* problem, we design the following process:

Algorithm 1 The Measurement of Arbitrarily Incorrect Label

$i \leftarrow n$
$d_v \leftarrow 0$
$d_r \leftarrow 0$
while $i > 0$ **do**
    $(X_1, y_1), (X_2, y_2), (X_3, y_3) \sim \mathcal{D}_{ABCD-Cog}$, where $y_1 < y_2 < y_3$ ▷ Randomly sample 3 data points and sorted by y
    $w = \frac{y_2 - y_3}{y_1 - y_3}$ ▷ Ensure $wy_1 + (1-w)y_3 = y_2$
    $X_{vmix} = wX_1 + (1-w)X_3$
    $X_{rmix} = \exp\left(w \log X_1 + (1-w) \log X_3\right)$
    $d_v += ||X_{vmix} - X_2||_1$
    $d_r += ||X_{rmix} - X_2||_1$
end while
$\overline{d_v} = \frac{d_v}{n}$
$\overline{d_r} = \frac{d_r}{n}$

We set n as 1000 and obtain $\overline{d_v} = 24,416.04 \pm 4,066.60$, $\overline{d_r} = 22,622.41 \pm 3,873.05$, where the sample distance $\overline{d_r}$ from R-Mixup is significantly smaller (7.3%) than the sample distance $\overline{d_v}$ from V-Mixup. The phenomenon shows our R-Mixup indeed can migrate the *arbitrarily incorrect label* problem.

Appendix D

Additional Information for Multi-task Learning Framework

D.1 Task Definition

D.2 Edge Importance and Task-level Correlation

The algorithm uses these symbols: $\mathcal{M}$ for the trained multi-task learning model, n_s for the number of samples selected from the test set, $\mathbf{X}^{(i)}$ and $\mathbf{Y}^{(i)}$ for the input brain networks and target labels of sample i, respectively, v for the number of views (e.g. rest-state or N-Back task), M for the number of nodes in each brain network, $\mathbf{IG}^{(i)}_{j,e,k}$ for the integrated gradients of edge e and task k in view j for sample i, t for the total number of tasks, $\mathbf{I\bar{G}}^{k}_{j}$ for the average edge importance of view j and task k, $\mathbf{g}^{k}_{j}$ for the vectorized edge importance of view j and task k, $\hat{\mathbf{G}}^{k}$ for the concatenated edge importance across views for task k, $\mathbf{G}^{k}$ for the averaged edge importance across views for task k, and $\mathbf{C}$ for the correlation matrix between task-level edge importance vectors.

Table D.1: Summary of tasks and their distributions. For numerical measures, the distribution is presented as mean±standard deviation.

Type	Task	ABCD field	Distribution
Cognition	Vocabulary	nihtbx_picvocab_uncorrected	85.29±7.73
	Attention	nihtbx_flanker_uncorrected	94.56±8.72
	Working Memory	nihtbx_list_uncorrected	97.69±11.47
	Executive Function	nihtbx_cardsort_uncorrected	93.37±8.86
	Processing Speed	nihtbx_pattern_uncorrected	88.75±14.36
	Episodic Memory	nihtbx_picture_uncorrected	103.58±12.03
	Reading	nihtbx_reading_uncorrected	91.40±6.52
	Fluid Cognition	nihtbx_fluidcomp_uncorrected	92.61±10.20
	Crystallized Cognition	nihtbx_cryst_uncorrected	87.10±6.65
	Overall Cognition	nihtbx_totalcomp_uncorrected	87.28±8.56
	Short Delay Recall	pea_ravlt_sd_trial_vi_tc	9.88±2.95
	Long Delay Recall	pea_ravlt_ld_trial_vii_tc	9.40±3.10
	Fluid Intelligence	pea_wiscv_trs	18.19±3.71
	Visuospatial Accuracy	lmt_scr_perc_correct	0.60±0.17
	Visuospatial Reaction time	lmt_scr_rt_correct	2691.27±461.01
Personality	Negative Urgency	upps_y_ss_negative_urgency	8.40±2.61
	Lack of Planning	upps_y_ss_lack_of_planning	7.68±2.29
	Sensation Seeking	upps_y_ss_sensation_seeking	9.84±2.67
	Positive Urgency	upps_y_ss_positive_urgency	7.86±2.89
	Lack of Perseverance	upps_y_ss_lack_of_perseverance	6.96±2.17
	Behavioral Inhibition	bis_y_ss_bis_sum	9.45±3.66
	Reward Responsiveness	bis_y_ss_bas_rr	10.94±2.90
	Drive	bis_y_ss_bas_drive	3.96±2.97
	Fun Seeking	bis_y_ss_bas_fs	5.65±2.59
Mental Health	Total Psychosis Symptoms	pps_y_ss_number	2.30±3.30
	Psychosis Severity	pps_y_ss_severity_score	5.33±9.44
	Anxious Depressed	cbcl_scr_syn_anxdep_r	2.45±3.01
	Withdrawn Depressed	cbcl_scr_syn_withdep_r	0.97±1.64
	Somatic Complaints	cbcl_scr_syn_somatic_r	1.46±1.92
	Social Problems	cbcl_scr_syn_social_r	1.46±2.13
	Thought Problems	cbcl_scr_syn_thought_r	1.53±2.08
	Attention Problems	cbcl_scr_syn_attention_r	2.71±3.30
	Rule-breaking Behavior	cbcl_scr_syn_rulebreak_r	1.07±1.70
	Aggressive Behavior	cbcl_scr_syn_aggressive_r	3.02±4.11
	Mania	pgbi_p_ss_score	1.16±2.56

Algorithm 2 Obtaining Task-level correlation matrix C and edge importance $\mathbf{G}^k$ for each task k by Integrated Gradients

Require: Trained multi-task learning model $\mathcal{M}$, test set $\mathcal{D}_{\text{test}}$, number of samples n_s

Ensure: Task-level correlation matrix C and edge importance $\mathbf{G}^k$ for each task k

Save the best-performing model $\mathcal{M}^*$ based on validation set performance
Randomly select n_s samples $\{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{n_s}$ from $\mathcal{D}_{\text{test}}$
for each sample $(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})$ **do**
    for each view $j \in \{1, \ldots, v\}$ **do**
        for each edge $e \in \{1, \ldots, M^2\}$ **do**
            for each task $k \in \{1, \ldots, t\}$ **do**
                Compute integrated gradients $\mathbf{IG}_{j,e,k}^{(i)}$ for edge e in view j and task k
            end for
        end for
    end for
end for
for each task $k \in \{1, \ldots, t\}$ **do**
    for each view $j \in \{1, \ldots, v\}$ **do**
        $\bar{\mathbf{IG}}_j^k = \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{IG}_{j,k}^{(i)}$ ▷ Average edge importance across samples
    end for
    $\hat{\mathbf{IG}}^k = \bigoplus_{j=1}^{v} \mathbf{g}_j^k$ ▷ Concatenate edge importance across views for task k
    $\mathbf{G}^k = \frac{1}{v} \sum_{j=1}^{v} \bar{\mathbf{IG}}_j^k$ ▷ Average IG across views for task k
end for
Compute correlation matrix $\mathbf{C} \in \mathbb{R}^{t \times t}$ between $\{\hat{\mathbf{IG}}^k\}_{k=1}^{t}$

Bibliography

[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*, page 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*, pages 1–15, 2019.

[4] Rushil Anirudh and Jayaraman J. Thiagarajan. Bootstrapping graph convolutional neural networks for autism spectrum disorder classification. In *ICASSP*, pages 3197–3201. IEEE, 2019.

[5] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proc. of ICML*, pages 233–242. PMLR, 2017.

[6] Sarp Aykent and Tian Xia. Gbpnet: Universal geometric representation learning on protein structures. In *KDD*, pages 4–14, 2022.

[7] 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.

[8] Guangji Bai and Liang Zhao. Saliency-regularized deep multi-task learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 15–25, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850.

[9] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Riemannian geometry applied to bci classification. In *International conference on latent variable analysis and signal separation*, pages 629–636. Springer, 2010.

[10] Deanna M Barch, Matthew D Albaugh, Shelli Avenevoli, Linda Chang, Duncan B Clark, Meyer D Glantz, James J Hudziak, Terry L Jernigan, Susan F Tapert, Debbie Yurgelun-Todd, et al. Demographic, physical and mental health assessments in the adolescent brain and cognitive development study: Rationale and description. *Developmental cognitive neuroscience*, 32:55–66, 2018.

[11] Danielle Smith Bassett and ED Bullmore. Small-world brain networks. *The neuroscientist*, 12(6):512–523, 2006.

[12] Nicole Berline, Ezra Getzler, and Micheèle Vergne. *Heat kernels and Dirac operators*. Grundlehren text editions. Springer, 2004. ISBN 9783540200628.

[13] Rajendra Bhatia, Stephane Gaubert, and Tanvi Jain. Matrix versions of the hellinger distance. *Letters in Mathematical Physics*, pages 1777–1804, 2019.

[14] Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the bures–wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 2019.

[15] Ronakben Bhavsar, Yi Sun, Na Helian, Neil Davey, David Mayor, and Tony Steffert. The correlation between eeg signals as measured in different positions on scalp varying with distance. *Procedia Computer Science*, pages 92–97, 2018.

[16] Peter J. Bickel and Bo Li. Local polynomial regression on unknown manifolds. *Lecture Notes-Monograph Series*, pages 177–186, 2007.

[17] Daniel A. Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Riemannian batch normalization for SPD neural networks. In *NeurIPS*, pages 15463–15474, 2019.

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

[19] 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.

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

[21] Eric Carlen. Trace inequalities and quantum entropy: an introductory course. In *Contemporary Mathematics*. American Mathematical Society, 2010. ISBN 9780821852477 9780821882085.

[22] 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*, pages 43–54, 2018.

[23] Rudrasis Chakraborty, Jose Bouza, Jonathan H. Manton, and Baba C. Vemuri. Manifoldnet: A deep neural network for manifold-valued data with applications. *TPAMI*, (2):799–810, 2022.

[24] C. Chefd'hotel, D. Tschumperlé, R. Deriche, and O. Faugeras. Regularizing Flows for Constrained Matrix-Valued Images. *Journal of Mathematical Imaging and Vision*, (1/2):147–162, 2004.

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

[26] Jiaao Chen, Zichao Yang, and Diyi Yang. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proc. of ACL*, pages 2147–2157. Association for Computational Linguistics, 2020.

[27] Jianzhong Chen, Angela Tam, et al. Shared and unique brain network features predict cognitive, personality, and mental health scores in the abcd study. *Nature communications*, 13(1):1–17, 2022.

[28] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: Fast learning with graph convolutional networks via importance sampling. In *Proc. of ICLR*. OpenReview.net, 2018.

[29] Junru Chen, Yang Yang, Tao Yu, Yingying Fan, Xiaolong Mo, and Carl Yang. Brainnet: Epileptic wave detection from seeg with hierarchical graph diffusion learning. In *KDD*, pages 2741–2751, 2022.

[30] Shi-Dong Chen, Jia You, Wei Zhang, Bang-Sheng Wu, Yi-Jun Ge, Shi-Tong Xiang, Jing Du, Kevin Kuo, Tobias Banaschewski, Gareth J Barker, et al. The genetic architecture of the human hypothalamus and its involvement in neuropsychiatric behaviours and disorders. *Nature Human Behaviour*, pages 1–15, 2024.

[31] Shuo Chen, Jian Kang, Yishi Xing, and Guoqing Wang. A parsimonious statistical method to detect groupwise differentially expressed functional connectivity networks. *Human brain mapping*, pages 5196–5206, 2015.

[32] Shuxiao Chen, Edgar Dobriban, and Jane H. Lee. A group-theoretic framework for data augmentation. In *NeurIPS*, 2020.

[33] Philip E. Cheng. Applications of kernel regression estimation:survey. *Communications in Statistics - Theory and Methods*, (11):4103–4134, 1990.

[34] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: rebalanced mixup. In *ECCV*, pages 95–110. Springer, 2020.

[35] 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*, pages 9355–9366, 2021.

[36] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, 2014.

[37] AO Cohen, MI Conley, DV Dellarco, and BJ Casey. The impact of emotional cues on short-term and long-term memory during adolescence. *Proceedings of the Society for Neuroscience. San Diego, CA. November*, 2016.

[38] 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*, pages 329–342, 1997.

[39] Marco Congedo and Alexandre Barachant. A special form of SPD covariance matrix for interpretation and visualization of data manipulated with Riemannian geometry. pages 495–503, 2015.

[40] Gheorghe Craciun and Martin Feinberg. Multiple equilibria in complex chemical reaction networks: Ii. the species-reaction graph. *SIAM Journal on Applied Mathematics*, (4):1321–1338, 2006.

[41] R Cameron Craddock, Paul E Holtzheimer III, Xiaoping P Hu, and Helen S Mayberg. Disease state prediction from resting state functional connectivity. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, pages 1619–1628, 2009.

[42] 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*, pages 1914–1928, 2012.

[43] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

[44] 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.

[45] 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.

[46] 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. *IEEE Transactions on Medical Imaging*, 42(2):493–506, 2023.

[47] Hejie Cui, Jiaying Lu, Shiyu Wang, Ran Xu, Wenjing Ma, Shaojun Yu, et al. A survey on knowledge graphs for healthcare: Resources, application progress, and promise. *arXiv*, 2023.

[48] Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, and Nasser M. Nasrabadi. Supermix: Supervising the mixing data augmentation. In *CVPR*, pages 13794–13803, 2021.

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

[50] Wei Dai, Hejie Cui, Xuan Kan, Ying Guo, and Carl Yang. Transformer-based hierarchical clustering for brain network analysis. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 4970–4971, 2022.

[51] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proc. of ACL*, pages 2978–2988. Association for Computational Linguistics, 2019.

[52] 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*, pages 43–56, 2011.

[53] Morris H DeGroot and Mark J Schervish. *Probability and statistics*. Pearson Education, 2012.

[54] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics, 2019.

[55] Luca Dodero, Hà Quang Minh, Marco San Biagio, Vittorio Murino, and Diego Sona. Kernel-based classification for brain connectivity graphs on the riemannian manifold of positive definite matrices. In *ISBI*, pages 42–45, 2015.

[56] 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 *Proc. of ICLR*. OpenReview.net, 2021.

[57] Ian L. Dryden, Alexey Koloydenko, and Diwei Zhou. Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *The Annals of Applied Statistics*, (3):1102 – 1123, 2009.

[58] Yuanqi Du, Shiyu Wang, Xiaojie Guo, Hengning Cao, Shujie Hu, Junji Jiang, Aishwarya Varala, Abhinav Angirekula, and Liang Zhao. Graphgt: Machine learning datasets for graph generation and transformation. In *NeurIPS*, 2021.

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

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

[61] Monique Ernst, Salvatore Torrisi, Nicholas Balderston, Christian Grillon, and Elizabeth A Hale. fmri functional connectivity applied to adolescent neurodevelopment. *Annual review of clinical psychology*, pages 361–377, 2015.

[62] Christian Feddern, Joachim Weickert, Bernhard Burgeth, and Martin Welk. Curvature-Driven PDE Methods for Matrix-Valued Images. *IJCV*, (1):93–107, 2006.

[63] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with

PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[64] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017.

[65] Emily S Finn, Xilin Shen, Dustin Scheinost, Monica D Rosenberg, Jessica Huang, Marvin M Chun, Xenophon Papademetris, and R Todd Constable. Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. *Nature neuroscience*, 18(11):1664–1671, 2015.

[66] Thomas Fletcher. Geodesic Regression on Riemannian Manifolds. In *Proceedings of the Third International Workshop on Mathematical Foundations of Computational Anatomy*, pages 75–86, 2011.

[67] Alex Fornito, Andrew Zalesky, and Michael Breakspear. Graph analysis of the human connectome: promise, progress, and pitfalls. *Neuroimage*, pages 426–444, 2013.

[68] Alex Fornito, Andrew Zalesky, and Michael Breakspear. The connectomics of brain disorders. *Nature Reviews Neuroscience*, pages 159–172, 2015.

[69] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *Proc. of ICML*, Proceedings of Machine Learning Research, pages 1972–1982. PMLR, 2019.

[70] KJ Friston, CD Frith, PF Liddle, and RSJ Frackowiak. Functional connectivity: the principal-component analysis of large (pet) data sets. *Journal of Cerebral Blood Flow & Metabolism*, pages 5–14, 1993.

[71] Soham Gadgil, Qingyu Zhao, Adolf Pfefferbaum, Edith V Sullivan, Ehsan Adeli, and Kilian M Pohl. Spatio-temporal graph convolution for resting-state fmri analysis. In *MICCAI*, pages 528–538. Springer, 2020.

[72] Jean Gallier and Jocelyn Quaintance. *Differential Geometry and Lie Groups: A Computational Perspective*. Geometry and Computing. Springer International Publishing, 2020.

[73] Giorgio Ganis and Stephen M. Kosslyn. Neuroimaging. In *Encyclopedia of the Human Brain*, pages 493–505. 2002.

[74] 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*, pages 105–124, 2013.

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

[76] Seppe Goovaerts, Hanne Hoskens, Ryan J Eller, Noah Herrick, Anthony M Musolf, Cristina M Justice, Meng Yuan, Sahin Naqvi, Myoung Keun Lee, Dirk Vandermeulen, et al. Joint multi-ancestry and admixed gwas reveals the complex genetics behind human cranial vault shape. *Nature communications*, 14 (1):7436, 2023.

[77] Michael D Greicius, Gaurav Srivastava, Allan L Reiss, and Vinod Menon. Default-mode network activity distinguishes alzheimer's disease from healthy aging: evidence from functional mri. *Proceedings of the National Academy of Sciences*, 101(13):4637–4642, 2004.

[78] 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*, pages 770–782, 2022.

[79] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *AAAI*, pages 3714–3722. AAAI Press, 2019.

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

[81] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pages 1024–1034, 2017.

[82] Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In *ICML*, pages 8230–8248. PMLR, 2022.

[83] Wolfgang Härdle. *Applied nonparametric regression*. Number 19. 1990.

[84] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[85] Wenchong He, Zhe Jiang, Chengming Zhang, and Arpan Man Sainju. Curvanet: Geometric deep learning based on directional curvature for 3d shape analysis. In *KDD*, pages 2214–2224. ACM, 2020.

[86] Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. Pixmix: Dreamlike pictures comprehensively improve safety measures. In *CVPR*, pages 16762–16771. IEEE, 2022.

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

[88] 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.

[89] 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.

[90] 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.

[91] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proc. of WWW*, pages 2704–2710. ACM / IW3C2, 2020.

[92] Chao Huang, Daniel Farewell, and Jianxin Pan. A calibration method for non-positive definite covariance matrix in multivariate data analysis. *Journal of Multivariate Analysis*, pages 45–52, 2017.

[93] Shuai Huang, James J. Lah, Jason W. Allen, and Deqiang Qiu. A probabilistic bayesian approach to recover r2* map and phase images for quantitative susceptibility mapping. *Magnetic Resonance in Medicine*, 88(4):1624–1642, 2022.

[94] Shuai Huang, James J Lah, Jason W Allen, and Deqiang Qiu. Robust quantitative susceptibility mapping via approximate message passing with parameter estimation. *Magnetic Resonance in Medicine*, 2023.

[95] Zhiwu Huang and Luc Van Gool. A riemannian network for SPD matrix learning. In *Proc. of AAAI*, pages 2036–2042. AAAI Press, 2017.

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

[97] Seong-Hyeon Hwang and Steven Euijong Whang. Mixrl: Data mixing augmentation for regression using reinforcement learning. *ArXiv preprint*, 2021.

[98] Seonghyeon Hwang and Steven Euijong Whang. Mixrl: Data mixing augmentation for regression using reinforcement learning. *CoRR*, 2021.

[99] Thomas R Insel and Bruce N Cuthbert. Brain disorders? precisely. *Science*, pages 499–500, 2015.

[100] Sadeep Jayasumana, Richard I. Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Tafazzoli Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *CVPR*, pages 73–80, 2013.

[101] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel Methods on Riemannian Manifolds with Gaussian RBF Kernels. *TPAMI*, (12):2464–2477, 2015.

[102] Junzhong Ji, Aixiao Zou, Jinduo Liu, Cuicui Yang, Xiaodan Zhang, and Yongduan Song. A survey on brain effective connectivity network learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(4):1879–1899, 2023.

[103] Yuelyu Ji, Yuhe Gao, Runxue Bao, Qi Li, Disheng Liu, Yiming Sun, and Ye Ye. Prediction of covid-19 patients' emergency room revisit using multi-source transfer learning. In *2023 IEEE 11th International Conference on Healthcare Informatics (ICHI)*, 2023.

[104] Mingxuan Ju, Shifu Hou, Yujie Fan, Jianan Zhao, Yanfang Ye, and Liang Zhao. Adaptive kernel graph neural network. In *AAAI*, pages 7051–7058, 2022.

[105] Xuan Kan, Hejie Cui, and Carl Yang. Zero-shot scene graph relation prediction through commonsense knowledge integration. In *Machine Learning and Knowledge Discovery in Databases.*, pages 466–482. Springer, 2021.

[106] Xuan Kan, Hejie Cui, Joshua Lukemire, Ying Guo, and Carl Yang. Fbnetgen: Task-aware gnn-based fmri analysis via functional brain network generation. In *MIDL*, pages 618–637. PMLR, 2022.

[107] Xuan Kan, Wei Dai, Hejie Cui, Zilong Zhang, Ying Guo, and Carl Yang. Brain network transformer. In *NeurIPS*, 2022.

[108] Xuan Kan, Aodong Chen Gu, Hejie Cui, Ying Guo, and Carl Yang. Dynamic brain transformer with multi-level attention for functional brain network analysis. 2023.

[109] Xuan Kan, Zimu Li, Hejie Cui, Yue Yu, Ran Xu, Shaojun Yu, Zilong Zhang, Ying Guo, and Carl Yang. R-mixup: Riemannian mixup for biological networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 1073–1085, New York, NY, USA, 2023. Association for Computing Machinery.

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

[111] 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*, pages 1038–1049, 2017.

[112] 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.

[113] Arielle S Keller, Adam R Pines, Sheila Shanmugan, Valerie J Sydnor, Zaixu Cui, Maxwell A Bertolero, Ran Barzilay, Aaron F Alexander-Bloch, Nora Byington, Andrew Chen, et al. Personalized functional brain network topography is associated with individual differences in youth cognition. *Nature communications*, 14(1):8411, 2023.

[114] Byung-Hoon Kim, Jong Chul Ye, and Jae-Jin Kim. Learning dynamic graph representation of brain connectome with spatio-temporal attention. In *NeurIPS*, pages 4314–4327, 2021.

[115] Hyunwoo J. Kim, Barbara B. Bendlin, Nagesh Adluru, Maxwell D. Collins, Moo K. Chung, Sterling C. Johnson, Richard J. Davidson, and Vikas Singh. Multivariate general linear models (MGLM) on riemannian manifolds with applications to statistical analysis of diffusion weighted images. In *CVPR*, pages 2705–2712. IEEE Computer Society, 2014.

[116] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *ICML*, pages 5275–5285. PMLR, 2020.

[117] Junghi Kim, Jeffrey R Wozniak, Bryon A Mueller, and Wei Pan. Testing group differences in brain functional connectivity: using correlations or partial correlations? *Brain connectivity*, pages 214–231, 2015.

[118] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR*. OpenReview.net, 2017.

[119] Serkan Kiranyaz, Turker Ince, Osama Abdeljaber, Onur Avci, and Moncef Gabbouj. 1-d convolutional neural networks for signal processing applications. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 8360–8364. IEEE, 2019.

[120] Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 21618–21629, 2021.

[121] Suprateek Kundu, Jin Ming, Jordan Pierce, Jennifer McDowell, and Ying Guo. Estimating dynamic brain functional networks using multi-subject fmri data. *NeuroImage*, pages 635–649, 2018.

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

[123] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995, 1995.

[124] John M. Lee. *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing, 2018.

[125] 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.

[126] 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.

[127] Zimeng Li, Shichao Zhu, Bin Shao, Xiangxiang Zeng, Tong Wang, and Tie-Yan Liu. Dsn-ddi: an accurate and generalized framework for drug–drug interaction prediction by dual-view representation learning. *Briefings in Bioinformatics*, (1), 2023.

[128] Qixiang Lin, Salman Shahid, Antoine Hone-Blanchet, Shuai Huang, Junjie Wu, Aditya Bisht, David Loring, Felicia Goldstein, Allan Levey, Bruce Crosson, et al. Magnetic resonance evidence of increased iron content in subcortical brain regions in asymptomatic alzheimer's disease. *Human Brain Mapping*, 2023.

[129] Sikun Lin, Shuyun Tang, Scott T Grafton, and Ambuj K Singh. Deep representations for time-varying brain datasets. In *KDD*, pages 999–1009, 2022.

[130] Martin A Lindquist. The statistical analysis of fmri data. *Statistical science*, pages 439–464, 2008.

[131] Lingwen Liu, Guangqi Wen, Peng Cao, Tianshun Hong, Jinzhu Yang, Xizhe Zhang, and Osmar R. Zaiane. Braintgl: A dynamic graph representation learning model for brain network analysis. *Computers in Biology and Medicine*, 153: 106521, 2023.

[132] Yong Liu, Meng Liang, Yuan Zhou, Yong He, Yihui Hao, Ming Song, Chunshui Yu, Haihong Liu, Zhening Liu, and Tianzi Jiang. Disrupted small-world networks in schizophrenia. *Brain*, pages 945–961, 2008.

[133] GD LOGAN. On the ability to inhibit thought and action: A users' guide to the stop signal paradigm. *Inhibitory processes in attention, memory, and language*, pages 189–239, 1994.

[134] Monica Luciana, James M Bjork, Bonnie J Nagel, Deanna M Barch, Raul Gonzalez, Sara Jo Nixon, and Marie T Banich. Adolescent neurocognitive development and impacts of substance use: Overview of the adolescent brain cognitive development (abcd) baseline neurocognition battery. *Developmental cognitive neuroscience*, 32:67–79, 2018.

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

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

[137] Sina Mansour L., Maria A. Di Biase, Robert E. Smith, Andrew Zalesky, and Caio Seguin. Connectomes for 40,000 uk biobank participants: A multi-modal, multi-scale brain network resource. *NeuroImage*, 283:120407, 2023. ISSN 1053-8119.

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

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

[140] Linghui Meng, Jin Xu, Xu Tan, Jindong Wang, Tao Qin, and Bo Xu. Mixspeech:

Data augmentation for low-resource automatic speech recognition. In *ICASSP 2021*, pages 7008–7012. IEEE, 2021.

[141] Paolo Mignone, Gianvito Pio, Sašo Džeroski, and Michelangelo Ceci. Multi-task learning for the simultaneous reconstruction of the human and mouse gene regulatory networks. *Scientific reports*, 10(1):22295, 2020.

[142] 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*, pages D412–D419, 2020.

[143] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, pages 5425–5434, 2017.

[144] Thomas E Nichols and Andrew P Holmes. Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human brain mapping*, pages 1–25, 2002.

[145] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 10th anniversary ed edition, 2010. ISBN 9781107002173.

[146] Paola Paci, Giulia Fiscon, Federica Conte, Rui-Sheng Wang, Lorenzo Farina, and Joseph Loscalzo. Gene co-expression in the interactome: moving from correlation toward causation via an integrated approach to disease module discovery. *NPJ systems biology and applications*, (1):1–11, 2021.

[147] Yue-Ting Pan, Jing-Lun Chou, and Chun-Shu Wei. Matt: A manifold attention network for eeg decoding. *NeurIPS*, 2022.

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

[149] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[150] Xavier Pennec. *Statistical Computing on Manifolds: From Riemannian Geometry to Computational Anatomy*, pages 347–386. Springer Berlin Heidelberg, 2009.

[151] Jonathan D Power, Alexander L Cohen, Steven M Nelson, Gagan S Wig, Kelly Anne Barnes, Jessica A Church, Alecia C Vogel, Timothy O Laumann, Fran M Miezin, Bradley L Schlaggar, et al. Functional network organization of the human brain. *Neuron*, 72(4):665–678, 2011.

[152] Jonathan D Power, Alexander L Cohen, Steven M Nelson, Gagan S Wig, Kelly Anne Barnes, Jessica A Church, Alecia C Vogel, Timothy O Laumann, Fran M Miezin, Bradley L Schlaggar, and Steven E Petersen. Functional Network Organization of the Human Brain. *Neuron*, pages 665–678, 2011.

[153] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[154] Daniel A. Roberts. *The principles of deep learning theory: an effective theory approach to understanding neural networks*. Cambridge University Press, 2022. ISBN 9781316519332.

[155] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *Proc. of ICLR*. OpenReview.net, 2020.

[156] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, pages 1059–1069, 2010.

[157] Jeffrey D Rudie, JA Brown, Devi Beck-Pancer, LM Hernandez, EL Dennis, PM Thompson, SY Bookheimer, and MJNC Dapretto. Altered functional and structural brain network organization in autism. *NeuroImage: clinical*, pages 79–94, 2013.

[158] Takashi Sakai. *Riemannian Manifolds*. Mathematical Monographs. 1996.

[159] Theodore D. Satterthwaite, Mark A. Elliott, Kosha Ruparel, James Loughead, Karthik Prabhakaran, Monica E. Calkins, Ryan Hopson, Chad Jackson, Jack Keefe, Marisa Riley, Frank D. Mentch, Patrick Sleiman, Ragini Verma, Christos Davatzikos, Hakon Hakonarson, Ruben C. Gur, and Raquel E. Gur. Neuroimaging of the Philadelphia Neurodevelopmental Cohort. *NeuroImage*, pages 544–553, 2014.

[160] 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*, pages 2383–2394, 2015.

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

[162] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[163] Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *Proc. of ICLR*. OpenReview.net, 2021.

[164] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004. ISBN 9780521813976.

[165] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. SGCN: sparse graph convolution network for pedestrian trajectory prediction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 8994–9003. Computer Vision Foundation / IEEE, 2021.

[166] 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.

[167] 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*, page 1, 2013.

[168] Stephen M Smith. The future of fmri connectivity. *Neuroimage*, pages 1257–1266, 2012.

[169] Stephen M Smith, Peter T Fox, Karla L Miller, David C Glahn, P Mickle Fox, Clare E Mackay, Nicola Filippini, Kate E Watkins, Roberto Toro, Angela R Laird, et al. Correspondence of the brain's functional architecture during activation and rest. *PNAS*, pages 13040–13045, 2009.

[170] 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*, 2011.

[171] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[172] Olaf Sporns. Structure and function of complex brain networks. *Dialogues in clinical neuroscience*, page 247, 2013.

[173] Cornelis J Stam, BF Jones, G Nolte, M Breakspear, and Ph Scheltens. Small-world networks and functional connectivity in alzheimer's disease. *Cerebral cortex*, pages 92–99, 2007.

[174] Yoon-Je Suh and Byung Hyung Kim. Riemannian embedding banks for common spatial patterns with eeg-based SPD neural networks. In *AAAI*, pages 854–862, 2021.

[175] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[176] Yann Thanwerdas and Xavier Pennec. O(n)-invariant Riemannian metrics on SPD matrices. *Linear Algebra and its Applications*, pages 163–201, 2023.

[177] C. G. Thomas, R. Harshman, and R. Menon. Noise reduction in bold-based fmri using component analysis. *NeuroImage*, pages 1521–1537, 2002.

[178] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar,

Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *ECCV*, 2022.

[179] Birkan Tunç, Berkan Solmaz, Drew Parker, Theodore D. Satterthwaite, Mark A. Elliott, Monica E. Calkins, Kosha Ruparel, Raquel E. Gur, Ruben C. Gur, and Ragini Verma. Establishing a link between sex-related differences in the structural connectome and behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, page 20150111, 2016.

[180] Timo Tuovinen, Riikka Rytty, Virpi Moilanen, Ahmed Abou Elseoud, Juha Veijola, Anne M. Remes, and Vesa J. Kiviniemi. The effect of gray matter ica and coefficient of variation mapping of bold data on the detection of functional connectivity changes in alzheimer's disease and bvftd. *Frontiers in Human Neuroscience*, page 680, 2017.

[181] David C Van Essen, Kamil Ugurbil, Edward Auerbach, Deanna Barch, Timothy EJ Behrens, Richard Bucholz, Acer Chang, Liyong Chen, Maurizio Corbetta, Sandra W Curtiss, et al. The human connectome project: a data acquisition perspective. *Neuroimage*, 62(4):2222–2231, 2012.

[182] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

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

[184] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *Proc. of ICLR*. OpenReview.net, 2019.

[185] Shashanka Venkataramanan, Ewa Kijak, Laurent Amsaleg, and Yannis Avrithis. Alignmixup: Improving representations by interpolating aligned features. In *CVPR*, pages 19174–19183, 2022.

[186] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *Proc. of ICML*, pages 6438–6447. PMLR, 2019.

[187] Shiyu Wang, Xiaojie Guo, Xuanyang Lin, Bo Pan, Yuanqi Du, Yinkai Wang, Yanfang Ye, Ashley Petersen, Austin Leitgeb, Saleh Alkhalifa, Kevin Minbiole, William M. Wuest, Amarda Shehu, and Liang Zhao. Multi-objective deep data generation with correlated property control. In *NeurIPS*, 2022.

[188] Shiyu Wang, Guangji Bai, Qingyang Zhu, Zhaohui Qin, and Liang Zhao. Domain generalization deep graph transformation, 2023.

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

[190] 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*, 2016.

[191] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Mixup for node and graph classification. In *the Web Conference*, pages 3663–3674, 2021.

[192] Leanne M Williams. Precision psychiatry: a neural circuit taxonomy for depression and anxiety. *The Lancet Psychiatry*, pages 472–480, 2016.

[193] Lirong Wu, Haitao Lin, Zhangyang Gao, Cheng Tan, Stan Li, et al. Graphmixup: Improving class-imbalanced node classification on graphs by self-supervised context prediction. *ArXiv preprint*, 2021.

[194] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proc. of ICML*, JMLR Workshop and Conference Proceedings, pages 478–487. JMLR.org, 2016.

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

[196] Ran Xu, Yue Yu, Chao Zhang, Mohammed K Ali, Joyce C Ho, and Carl Yang. Counterfactual and factual reasoning over hypergraphs for interpretable clinical predictions on ehr. In *Machine Learning for Health*, pages 259–278. PMLR, 2022.

[197] Ran Xu, Yue Yu, Hejie Cui, Xuan Kan, Yanqiao Zhu, Joyce Ho, Chao Zhang, and Carl Yang. Neighborhood-regularized self-training for learning with few labels. In *AAAI Conference on Artificial Intelligence*, 2023.

[198] Ran Xu, Yue Yu, Joyce C Ho, and Carl Yang. Weakly-supervised scientific document classification via retrieval-augmented multi-stage training. In *SIGIR*, 2023.

[199] Noriaki Yahata, Jun Morimoto, Ryuichiro Hashimoto, Giuseppe Lisi, Kazuhisa Shibata, Yuki Kawakubo, Hitoshi Kuwabara, Miho Kuroda, Takashi Yamada, Fukuda Megumi, Hiroshi Imamizu, José E. Náñez Sr, Hidehiko Takahashi, Yasumasa Okamoto, Kiyoto Kasai, Nobumasa Kato, Yuka Sasaki, Takeo Watanabe, and Mitsuo Kawato. A small number of abnormal brain connections predicts adult autism spectrum disorder. *Nature Communications*, page 11254, 2016.

[200] 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.

[201] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. Heterogeneous network representation learning: A unified framework with survey and benchmark. *TKDE*, 2020.

[202] 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.

[203] 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. In *KDD '22*, page 4743–4751, New York, NY, USA, 2022.

[204] Yi Yang, Hejie Cui, and Carl Yang. Ptgb: Pre-train graph neural networks for brain network analysis. In *The Conference on Health, Inference, and Learning*, 2023.

[205] Huaxiu Yao, Yiping Wang, Linjun Zhang, James Zou, and Chelsea Finn. C-mixup: Improving generalization in regression. In *NeurIPS*, 2022.

[206] Özgür Yeniay. Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and Computational Applications*, 2005.

[207] 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*, pages 28877–28888, 2021.

[208] Kisung You and Hae-Jeong Park. Geometric learning of functional brain network on the correlation manifold. *Scientific Reports*, pages 1–13, 2022.

[209] Haiyuan Yu, Philip M Kim, Emmett Sprecher, Valery Trifonov, and Mark Gerstein. The importance of bottlenecks in protein networks: correlation with gene essentiality and expression dynamics. *PLoS computational biology*, (4), 2007.

[210] Yue Yu, Kexin Huang, Chao Zhang, Lucas M Glass, Jimeng Sun, and Cao Xiao. Sumgnn: multi-typed drug interaction prediction via efficient knowledge graph summarization. *Bioinformatics*, (18):2988–2995, 2021.

[211] Yue Yu, Xuan Kan, Hejie Cui, Ran Xu, Yujia Zheng, Xiangchen Song, Yanqiao Zhu, Kun Zhang, et al. Learning task-aware effective brain connectivity for fmri analysis with graph neural networks. *ArXiv preprint*, 2022.

[212] Yue Yu, Rongzhi Zhang, Ran Xu, Jieyu Zhang, Jiaming Shen, and Chao Zhang. Cold-start data selection for few-shot language model fine-tuning: A prompt-based uncertainty propagation approach. *ArXiv preprint*, 2022.

[213] Yue Yu, Xuan Kan, Hejie Cui, Ran Xu, Yujia Zheng, Xiangchen Song, Yanqiao Zhu, Kun Zhang, Razieh Nabi, Ying Guo, Chao Zhang, and Carl Yang. Deep dag learning of effective brain connectivity for fmri analysis. In *ISBI*, 2023.

[214] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6022–6031. IEEE, 2019.

[215] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

[216] Andrew Zalesky, Alex Fornito, and Edward T Bullmore. Network-based statistic: identifying differences in brain networks. *Neuroimage*, pages 1197–1207, 2010.

[217] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proc. of ICLR*. OpenReview.net, 2018.

[218] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5171–5181, 2018.

[219] Muhan Zhang and Pan Li. Nested graph neural networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15734–15747, 2021.

[220] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proc. of AAAI*, pages 4438–4445. AAAI Press, 2018.

[221] Rongzhi Zhang, Yue Yu, and Chao Zhang. SeqMix: Augmenting active sequence labeling via sequence mixup. In *Proc. of EMNLP*, pages 8566–8579. Association for Computational Linguistics, 2020.

[222] Rongzhi Zhang, Yue Yu, Jiaming Shen, Xiquan Cui, and Chao Zhang. Local boosting for weakly-supervised learning. In *KDD*, 2023.

[223] Shaofeng Zhang, Meng Liu, Junchi Yan, Hengrui Zhang, Lingxiao Huang, Xiaokang Yang, and Pinyan Lu. M-mix: Generating hard negatives via multi-sample mixing for contrastive learning. In *KDD*, pages 2461–2470, 2022.

[224] Yanfu Zhang and Heng Huang. New graph-blind convolutional network for brain connectome data analysis. In *IPMI*, 2019.

[225] 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.

[226] 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.