Brain Network Analysis by Graph Representation Learning

XU Jiaxing

College of Computing and Data Science

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy (Ph.D)

2025

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

15/08/2024
.....
Date

.....
Xu Jiaxing

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

16/08/2024
.....
Date

.....
Ke Yiping, Kelly

Authorship Attribution Statement

This thesis contains material from five papers published in the following peer-reviewed journals in which I am listed as an author.

Chapter 4 is published as **Jiaxing Xu***, Yunhan Yang*, David Tse Jung Huang*, Sophi Shilpa Gururajapathy*, Yiping Ke, Miao Qiao, Alan Wang, Haribalan Kumar, Josh McGeown, and Eryn Kwon. "Data-Driven Network Neuroscience: On Data Collection and Benchmark." in *Proceedings of the 37th Conference on Neural Information Processing Systems*, 2023.

The contributions of the co-authors are as follows:

- I developed the coding framework, performed the majority of the experiments and revised the manuscript.
- Yunhan Yang, David Tse Jung Huang, and Sophi Shilpa Gururajapathy performed the literature review, curated the majority of the datasets, and prepared the manuscript drafts.
- Prof. Yiping Ke and Prof. Miao Qiao provided the initial project direction, co-supervised the project, and revised the manuscript.
- Alan Wang, Haribalan Kumar, Josh McGeown and Eryn Kwon co-curated part of the datasets, revised the code, and revised the manuscript.

Chapter 5 is published as **Jiaxing Xu**, Jinjie Ni, and Yiping Ke. "A Class-Aware Representation Refinement Framework for Graph Classification." in *Information Sciences*, 2024.

The contributions of the co-authors are as follows:

- I co-developed the idea of the project, performed the literature review, developed the codebase, performed the experiments, and prepared the manuscript.
- Jinjie Ni co-developed the idea of the project and revised the manuscript drafts.
- Prof. Yiping Ke supervised the project and revised the manuscript drafts.

Chapter 6 is published as **Jiaxing Xu**, Qingtian Bian, Xinhang Li, Aihu Zhang, Yiping Ke, Miao Qiao, Wei Zhang, Wei Khang Jeremy Sim, and Balázs Gulyás. "Contrastive Graph Pooling for Explainable Classification of Brain Networks." in *IEEE Transactions on Medical Imaging*, 2024.

The contributions of the co-authors are as follows:

- I co-developed the idea of the project, performed the literature review, developed the codebase, performed the experiments, and prepared the manuscript.
- Qingtian Bian, Xinhang Li and Aihu Zhang co-developed the idea of the project, revised the code, and revised the manuscript drafts.
- Prof. Yiping Ke and Prof. Miao Qiao co-supervised the project, and revised the manuscript.
- Wei Zhang, Wei Khang Jeremy Sim and Prof. Balázs Gulyás provided feedback and revised the manuscript.

Chapter 7 is published as **Jiaxing Xu**, Kai He, Mengcheng Lan, Qingtian Bian, Wei Li, Tieying Li, Yiping Ke, and Miao Qiao. "Contrasformer: A Brain Network Contrastive Transformer for Neurodegenerative Condition Identification." in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024.

The contributions of the co-authors are as follows:

- I co-developed the idea of the project, performed the literature review, developed the codebase, performed the experiments, and prepared the manuscript.
- Kai He and Mengcheng Lan co-developed the idea of the project, and revised the manuscript drafts.
- Qingtian Bian, Wei Li and Tieying Li revised the code and provided feedback.
- Prof. Yiping Ke and Prof. Miao Qiao co-supervised the project, and revised the manuscript.

Chapter 8 is from **Jiaxing Xu**, Mengcheng Lan, Xia Dong, Kai He, Wei Zhang, Qingtian Bian, and Yiping Ke. "Multi-Atlas Brain Network Classification through Consistency Distillation and Complementary Information Fusion". The manuscript is still *Under Review*.

The contributions of the co-authors are as follows:

- I co-developed the idea of the project, performed the literature review, developed the codebase, performed the experiments, and prepared the manuscript.
- Mengcheng Lan and Xia Dong co-developed the idea of the project, and revised the manuscript drafts.
- Kai He, Wei Zhang and Qingtian Bian revised the code and provided feedback.
- Prof. Yiping Ke supervised the project, and revised the manuscript.

15/08/2024 .....
Date

..... Xu Jiaxing

Acknowledgments

I extend my deepest gratitude to my supervisor, Prof. Yiping Ke, whose unwavering guidance, expertise, and encouragement have been invaluable throughout my Ph.D. journey. Her consistent support and insightful discussions have been instrumental in my academic growth.

I also wish to express my appreciation to the esteemed members of my thesis advisory committee, Prof. Miao Qiao and Prof. Cheng Long, for their valuable suggestions and unwavering support, which have enriched the quality of my work.

My heartfelt thanks go to the dedicated collaborators who have contributed to this endeavor. I extend my gratitude to Qingtian Bian, Aihu Zhang, Mengcheng Lan, Wei Li, Han Lei, Xia Dong, Sophi Shilpa Gururajapathy, Tieying Li, Kai He, Jinjie Ni, Vijay Prakash Dwivedi, Xinhang Li, Yunhan Yang, David Tse Jung Huang, Alan Wang, Haribalan Kumar, Josh McGeown, Eryn Kwon, Wei Zhang, Wei Khang Jeremy Sim, and Balázs Gulyás. Their collaboration has been instrumental in the success of this project.

I would also like to acknowledge the cherished friendships that have accompanied me on this journey. To Haiyu Feng, Jiangneng Li, Xiaobei Yan, Yongqiang Chen, Qika Lin, Weihang Xie, Jingwei Zhou, Wei Shao, Jianhong Li, Bangtai Zhou, Junkai Zhang, Xiaoming Chen, Zizheng Que, and many more friends, I am grateful for the shared experiences and support that have made this academic pursuit more rewarding.

Mentoring undergraduate students for their final year projects has also provided me with valuable insights on how to lead and supervise technical and research projects. I would like to thank Ruotong Yu, Jacintha Wee Yun Yi, Ang Ser Gin Marcus, and Yee Kai Wen for the opportunity to work with them. I have learned much from our collaborations and hope they have gained useful skills and knowledge from this research experience.

Finally, my heartfelt thanks go to my parents for their unconditional love and unwavering support. It is their love and encouragement that have provided me with the strength to overcome even the most challenging moments in life.

Abstract

Brain networks, constructed from functional Magnetic Resonance Imaging (fMRI) data, play a crucial role in understanding the neural basis of various neurological disorders. However, analyzing these networks presents several challenges, including the complexities of data preprocessing, the limitations of traditional Graph Neural Networks (GNNs) in capturing graph-level relationships, and the inconsistencies arising from using multiple brain atlases.

This thesis first addresses the need for high-quality brain network data by compiling a comprehensive collection of functional human brain networks. By overcoming domain-specific preprocessing hurdles and computational demands, these datasets, consisting of fMRI data from 2,702 subjects across various sources and conditions, facilitate further research in neuroscience and machine learning.

Second, the thesis tackles the limitations of traditional GNNs in graph classification tasks. Traditional GNNs often neglect graph-level relationships and suffer from generalization issues due to treating each graph independently during message passing and pooling. To address these issues, this work introduces several advanced GNN frameworks designed to enhance class separability and generalization capabilities. These frameworks integrate robust class representations and leverage innovative attention mechanisms to steer the learning process, achieving superior performance in brain network classification tasks and aligning extracted patterns with domain knowledge.

Finally, the thesis addresses the challenges of utilizing multiple brain atlases for network classification. The lack of consistency and information exchange between different atlases hampers the detection of abnormalities in brain networks. This research introduces a novel approach that employs a disentangle mechanism to filter inconsistent information and fuse distinguishable connections across atlases. This ensures subject- and population-level consistency, significantly improving classification accuracy and efficiency.

Overall, this thesis advances the field of brain network analysis through innovative graph representation learning techniques. By providing high-quality datasets and developing advanced GNN frameworks, it offers valuable insights into neurological disorders and enhances the accuracy and interpretability of brain network classification.

Contents

List of Figures

List of Tables

List of Abbreviations

ABIDE	Autism Brain Imaging Data Exchange
AD	Alzheimer's Disease
ADNI	Alzheimer's Disease Neuroimaging Initiative
AFNI	Analysis of Functional NeuroImages
ANT	Advanced Normalization Tools
ASD	Autism Spectrum Disorder
BOLD	blood-oxygen-level-dependent
CARE	Class-Aware Representation rEfinement framework
CDA	Contrastive Dual-Attention
CN	cognitive normal
Contrasformer	Contrastive Brain Network Transformer
DTI	Diffusion Tensor Imaging
fMRI	functional Magnetic Resonance Imaging
GCN	Graph Convolution Network
GNN	Graph Neural Network
INDI	International Neuroimaging Data-sharing Initiative
kNN	k-Nearest Neighbours
LR	Logistic Regression
MCI	mild cognitive impairment
ML	Machine Learning
MLP	multilayer perceptron
NB	Gaussian Naive Bayes
PCP	Preprocessed Connectomes Project
PD	Parkinson's disease
PET	positron emission tomography
RF	Random Forest
ROI	region of interest
SMC	significant memory concern
SVC	Support Vector Machine Classifier
SWEDD	scans without evidence of dopaminergic deficit
TC	Typical Control
T1w	T1-weighted

List of Notations

$\boldsymbol{M}$	Connectivity matrix of a subject
G	An input graph/brain network
$\boldsymbol{A}$	Adjacency matrix of G
$\boldsymbol{X}$	Node feature matrix of G
$\mathcal{V}_G$	Node set of G
v, u	A node in G
$\boldsymbol{H}$	Node representations
$\boldsymbol{H}_v$	Node representation of v
m	Number of nodes in G
$\mathcal{D}$	Input dataset
$\mathcal{G}$	Input graph set
$\mathcal{Y}$	Input label set
y_G	Label of G
l	The layer index in GNN
d	Dimensionality of node representation $\boldsymbol{H}_v$
$\mathcal{G}^{TC}, \mathcal{G}^{ASD}$	Graph set of TC/ASD group
$\mathcal{A}^{TC}, \mathcal{A}^{ASD}$	Adjacency matrix set of TC/ASD group
$\mathcal{X}^{TC}, \mathcal{X}^{ASD}$	Feature matrix set of TC/ASD group
$G^{TC}_{sum}, G^{ASD}_{sum}$	Summary graph of TC/ASD group
$\boldsymbol{A}^{TC}_{sum}, \boldsymbol{A}^{ASD}_{sum}$	Summary adjacency matrix of TC/ASD group
$\boldsymbol{X}^{TC}_{sum}, \boldsymbol{X}^{ASD}_{sum}$	Summary feature matrix of TC/ASD group
$G_{contrast}$	Contrast graph
$\boldsymbol{A}_{contrast}$	Adjacency matrix of $G_{contrast}$
$\boldsymbol{H}_{contrast}$	Node representations of $G_{contrast}$
$\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V$	Parameter matrix
i, j	Index for matrix dimensions
n^{TC}	Number of subjects in TC group
$\boldsymbol{T}^{TC}_{ROI}$	Output 3D matrix of $\text{Attn}_{ROI}(\cdot)$ for TC group
$\boldsymbol{T}^{TC}_{subject}$	Output 3D matrix of $\text{Attn}_{subject}(\cdot)$ for TC group
$\boldsymbol{S}$	Cluster assignment matrix
$\boldsymbol{Z}$	Embedded node feature matrix
$\boldsymbol{A}_{contrast}(i, :)$	The i-th row of $\boldsymbol{A}_{contrast}$

Chapter 1

Introduction

1.1 Background

The field of neuroscience has made significant strides in unveiling the principles and mechanisms that underlie complex brain functions, both in individuals with normal brain function and those with neurological, psychiatric, and/or neurodevelopmental conditions. In recent years, there has been a notable expansion in the size, scope, and complexity of human neural data acquisition. This expansion, coupled with rapid advancements in machine learning and graph analytics, has led to a growing interest in the burgeoning discipline known as network neuroscience [3]. Network neuroscience focuses on understanding the structure and function of the human brain using graphs, often referred to as brain networks.

1.1.1 Brain Network Construction

The advent of neuroimaging has revolutionized our understanding of the human brain, offering unprecedented insights into its structure and function. Among the various brain imaging modalities available, each varies in invasiveness—from ex-vivo studies using extracted tissue samples to in-vivo studies conducted with scanners—and in image resolution, encompassing both spatial and temporal aspects. This thesis focuses on whole-brain imaging using functional magnetic resonance imaging (fMRI), which provides an optimal balance of spatial and temporal resolution while covering the entire brain. The fMRI data monitors changes in the blood flow, i.e., blood-oxygen-level-dependent (BOLD) signals to capture functional activities [4]. In particular, fMRI has been instrumental in identifying underlying neurodegenerative conditions, including Alzheimer's, Parkinson's, and Autism [5].

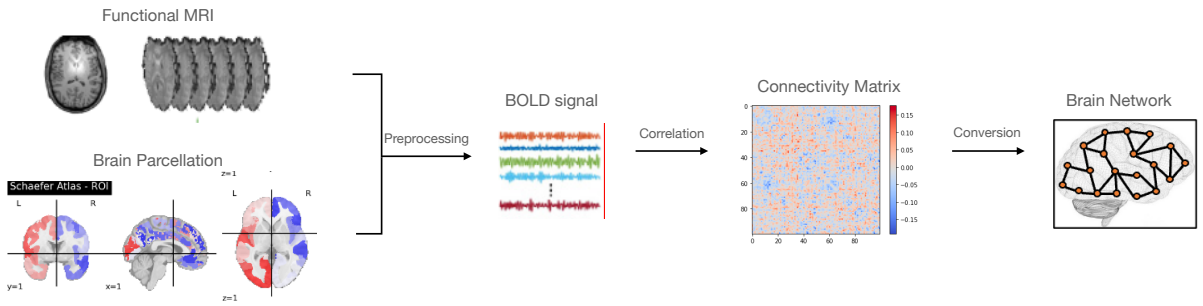


Fig. 1.1: Preprocessing steps for brain network.

The conversion of fMRI scans to brain networks has two stages, preprocessing and brain network construction, both requiring intensive domain inputs. Specifically, to ensure the image quality for subsequent tasks, the preprocessing of raw MRI images needs proper quality control over a number of steps, including motion correction, realigning, field unwarping, normalization, bias field correction, and brain extraction. Different preprocessing choices lead to a large variation in the output images. Parcellation translates the preprocessed MRI images to regions of interest (ROIs) as nodes and the co-activation between ROIs as weighted edges [6]. Each generated brain network contains i) a weighted adjacency matrix that characterizes the connectivity between ROIs and ii) a feature matrix that captures the attributes of ROIs in terms of the aggregated BOLD signals. Choosing a different brain atlas/scheme for parcellation leads to a different brain network. The preprocessing steps for brain networks are shown in Fig. 1.1.

1.1.2 Brain Network Analysis with Graph Representation Learning

With the ubiquity of graph-structured data emerging from various modern applications, Graph Neural Networks (GNNs) have gained increasing attention from both researchers and practitioners. GNNs have been applied to many application domains, including quantum chemistry [7–9], social science [10–12], transportation [13, 14] and neuroscience [15, 16], and have attained promising results on graph classification [9, 10], node classification [17] and link prediction [18, 19] tasks. Fig. 1.2, shows some examples of domains with graph-structured data.

Graph analytics methods are also applied to these brain networks to gain insights into the elements and interactions of neurological systems from a network perspective. While previous research has demonstrated the potential of network-based approaches [2, 20] in deciphering

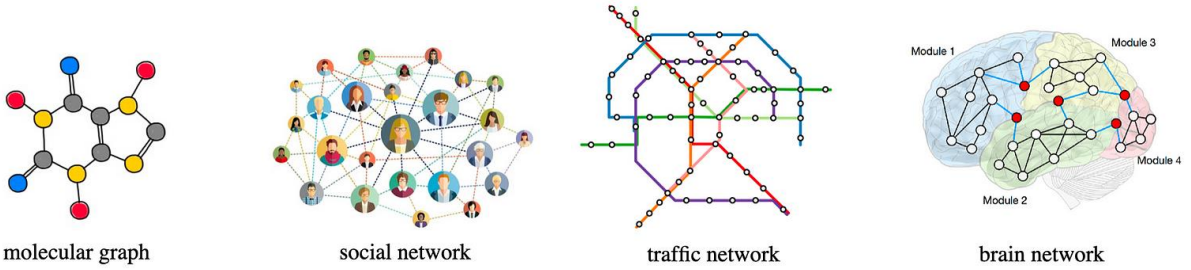


Fig. 1.2: Some example domains with graph-structured data.

the complexities of the brain, the field of graph analytics for brain networks is still in its early stages of development.

Network neuroscience leverages graph-based machine learning techniques for a range of clinically important applications. For instance, community detection within brain networks can reveal areas of co-activation that may be weakened in individuals with neurodegenerative conditions [2]. Techniques such as graph classification [21] have proven valuable in distinguishing subjects with neurological diseases from healthy individuals, while graph ordinal regression [22] can help identify individuals at different stages of neurological diseases based on the severity of their condition.

Unlike conventional machine learning models that primarily handle vector-based data, GNNs have the capacity to incorporate graph topological information through message passing. Given their promising performance in diverse applications, several studies have extended the use of GNNs to the analysis of brain networks [23–25].

1.2 Motivations

The conversion of fMRI scans into brain networks involves two crucial stages: preprocessing and brain network construction, both demanding specialized domain knowledge and tools. Preprocessing of raw MRI images is vital to ensure image quality for subsequent tasks, encompassing steps like motion correction, realignment, field unwarping, normalization, bias field correction, and brain extraction. Choices made during preprocessing significantly influence the output images. Parcellation further translates the preprocessed fMRI images into ROIs as nodes and represents co-activation between ROIs as weighted edges. Each resulting brain network consists of a weighted adjacency matrix characterizing ROI connectivity and a feature

matrix capturing ROI attributes based on aggregated BOLD signals. The selection of a specific brain atlas or parcellation scheme directly impacts the structure of the brain network. While existing studies commonly employ a single parcellation scheme to create a fixed set of nodes for all subjects, the impact of varying group-wise, data-driven parcellation schemes remains underexplored. This conversion process poses significant barriers to entry for research in brain networks, limiting the development of data-driven network neuroscience. It necessitates domain expertise for selecting preprocessing pipelines, entails high computational costs for image processing and graph extraction, and becomes complex when conducting large-scale imaging studies with diverse equipment and acquisition protocols. In light of these challenges, this thesis strives to bridge this gap by making a comprehensive collection of brain networks available to the public. We anticipate that the release of this brain network dataset will promote interdisciplinary research in network neuroscience, machine learning, and graph analytics, ultimately advancing studies that employ graph-based techniques for the detection of neurodegenerative conditions.

However, the direct application of general-purpose GNNs to fMRI data faces certain challenges due to the unique characteristics of this data [26]. First, fMRI data typically exhibit a low signal-to-noise ratio, resulting from non-neural noise sources such as cardiac and respiratory processes or scanner instability, leading to substantial variations within and across subjects. Second, the nodes in a brain network correspond to ROIs under a specific parcellation scheme, resulting in a consistent number of nodes and alignment across different subjects. Lastly, due to limited availability, brain network datasets often contain a relatively small number of subjects, which can lead to overfitting when employing GNNs.

Moreover, most GNNs exhibit two primary limitations when used for downstream classification tasks. They often overlook graph-level relationships, treating input graphs independently in their training processes and neglecting potential similarities or discrepancies among different graphs. Additionally, GNNs may face generalization issues, particularly when networks are deep or have high hidden dimensionality. Several methods have been proposed to address these generalization challenges, including graph augmentation, adversarial learning, and resampling, but many of these techniques focus on individual graphs and fail to effectively leverage graph-level information to improve generalization.

In addition, brain network construction involves using a specific atlas to parcellate the brain into ROIs. Various atlases based on different hypotheses of brain parcellation, such as anatomical and functional divisions, have been proposed to group similar fMRI regions and create ROIs [27–29]. Although proper brain parcellation is essential for detecting abnormalities in neurodegenerative disorders [30], there is no golden standard atlas for brain network classification. Relying on a single atlas for brain network analysis has two main drawbacks. First, some voxels may not be assigned to any specific ROI, potentially leading to the loss of important information. Second, each atlas is based on a different parcellation hypothesis. The BOLD signal of an ROI is averaged from all voxels within it, possibly missing detailed information. To address these limitations, recent works have proposed using multiple atlases with different parcellation modes to enhance multi-atlas brain network analysis. Some methods [31, 32] independently encode brain networks from various atlases and then aggregate the graph representations as a late feature fusion scheme for the final prediction. Another approach [33] incorporates early feature fusion by incorporating multi-atlas information from the raw data and using the fused feature for representation learning. However, these methods (1) neglect the need of consistency across atlases, potentially leading to the under-utilization of cross-atlas information; and (2) lack ROI-level information exchange throughout the entire representation learning process, which could hinder the models' ability to discern complementary information across different atlases.

1.3 Contributions

As evident from the preceding discussions, effective brain analysis requires the integration of domain knowledge in constructing brain networks and the consideration of specific fMRI characteristics in model design. To this end, our work pertaining to suitable dataset construction and methodology designs are summarized below. Fig. 1.3 provides an overview of the challenges, solutions, and targets of each part.

1.3.1 Datasets and Benchmarks

- This thesis releases a large resting-state functional brain network collection to the public. The collection was originated from 6 raw rs-fMRI image sources with 5 well recognized

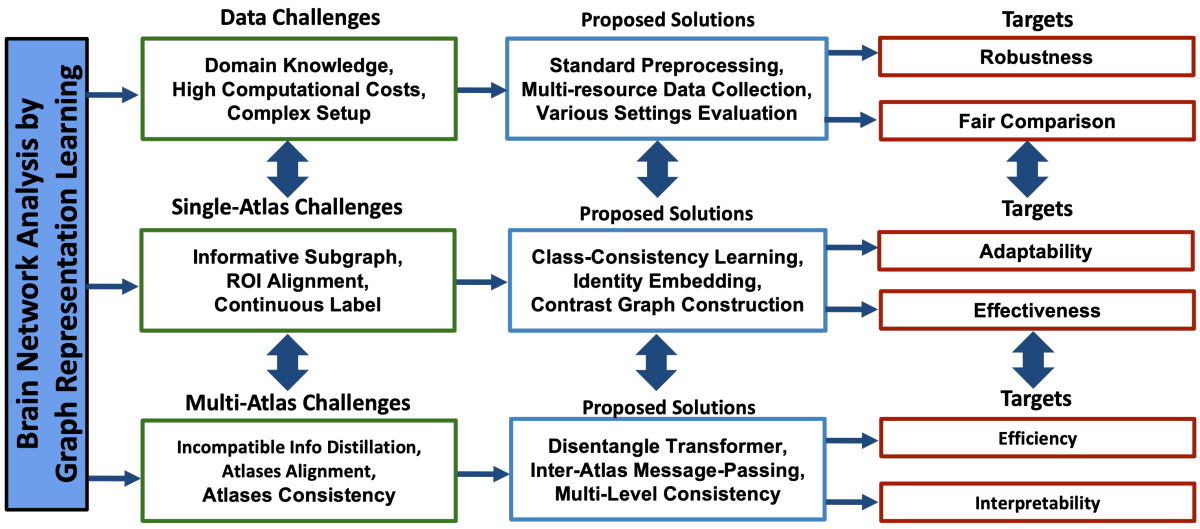


Fig. 1.3: The summary of the challenges this thesis focused on, the proposed solutions, and the final targets we achieved.

ones in neuroscience and one new data source, covering 3 neurodegenerative and one brain injury conditions, i.e., Autism, Alzheimer, Parkinson, and mTBI. The collection consists of ABIDE (N=1025), ADNI (N=1327), PPMI (N=209), Mātai (N=60) and two other sources totalling to 2,702 subjects.

- This thesis tests the datasets on one recent graph analysis model [2], 6 conventional machine learning (ML) models, as well as 6 representative graph ML models. The experimental results demonstrate that the quality of our datasets is not compromised by the conversion process and can serve as a domain benchmark for subsequent research.

1.3.2 Class-Aware Representation Refinement Framework

- This thesis proposes a novel graph representation refinement framework CARE, which considers class-aware graph-level relationships. CARE is a flexible plug-and-play framework that can incorporate arbitrary GNNs without significantly increasing the computational cost.

- This thesis provides theoretic support through VC dimension analysis that CARE has a better generalization upper bound in comparison with its GNN backbone.

- This thesis performs extensive experiments using 11 GNN backbones on 14 benchmark datasets to justify the superiority of CARE on graph classification task in terms of both effectiveness and efficiency.

1.3.3 Contrastive Graph Pooling

- This thesis proposes a contrastive dual-attention block, which adaptively assigns a weight to each ROI of each subject. By aggregating subjects in each group, this thesis introduces a differentiable graph pooling method called *ContrastPool* to select the most discriminative regions w.r.t different groups.

- This thesis applies our ContrastPool to 5 resting-state fMRI brain network datasets spanning over 3 diseases. The results justify the superiority of ContrastPool over the state-of-the-art baselines.

- The case study confirms the interestingness, simplicity and high explainability of the patterns extracted by our method, which match the domain knowledge in neuroscience literature.

1.3.4 Contrastive Transformer

- This chapter introduces *Contrasformer*, a contrastive brain network Transformer, which dynamically assigns weights across ROIs and subjects to generate a contrast graph. This contrast graph encapsulates the most discriminative regions concerning different groups, facilitating its integration into graph representation learning for downstream tasks.

- This chapter evaluates Contrasformer on four resting-state fMRI brain network datasets featuring different neurodegenerative disorders. Our results demonstrate the superiority of Contrasformer over state-of-the-art baseline methods on neurodegenerative condition identification.

- This chapter presents a case study that underscores the intriguing, straightforward, and highly interpretable patterns extracted by our approach, aligning with domain knowledge found in neuroscience literature.

1.3.5 Atlas-Integrated Distillation and Fusion Network

- This chapter proposes a multi-atlas solution for brain network classification with fMRI data. AIDFusion takes full advantage of multi-atlas brain networks by enhanced atlas-consistent information distillation and intense fusion of cross-atlas complementary information.

- This chapter evaluates AIDFusion on four resting-state fMRI brain network datasets for different neurological disorders. Our results demonstrate the superiority of AIDFusion over state-of-the-art baseline methods in terms of effectiveness and efficiency in brain network classification.

- This chapter presents a case study that underscores the intriguing, straightforward, and highly interpretable patterns extracted by our approach, aligning with domain knowledge found in neuroscience literature.

1.4 Outline of the Thesis

This section provides a brief overview of the thesis structure and presents an outline of the following chapters.

Chapter 2 exhaustively reviews the existing literature about brain networks and GNNs that are related to our work. Chapter 3 specifies the symbols and formulas to be used in this thesis.

The remaining chapters are organized in three parts: **Part I** focuses on dataset construction and benchmark, **Part II** on single-atlas brain network analysis, and **Part III** on multi-atlas brain network analysis.

Part I: Dataset Construction and Benchmark for Brain Networks

Chapter 4 provides the details of our data collection, preprocessing and brain network construction.

Part II: Single-atlas Brain Network Analysis

Chapter 5 introduces our first work about the Class-Aware Representation Refinement (CARE) Framework.

Chapter 6 introduces our second work about the Contrastive Graph Pooling (ContrastPool).

Chapter 7 introduces our third work about the Contrastive Transformer (Contrasformer).

Part III: Multi-atlas Brain Network Analysis

Chapter 8 introduces our fourth work about the Atlas-Integrated Distillation and Fusion network (AIDFusion).

Finally, this thesis summarizes and conclude in Chapter 9, where this thesis also discusses the limitations and potential future work that can arise based on the insights presented in this thesis.

Chapter 2

Literature Review

In this chapter, we present a review of the field of brain network analysis and graph representation learning, with a focus on deep learning methods for graphs.

2.1 Data Collection and Benchmark

2.1.1 Data Collection.

Preprocessed Connectomes Project (PCP) [34] is an initiative to preprocess part of the raw MRI images in the International Neuroimaging Data-sharing Initiative (INDI) database and make the preprocessed neuroimages publicly available. Within PCP, one relevant dataset that has gone through functional preprocessing pipelines is the *ABIDE dataset* on Autism. Note that the pipeline used for preprocessing ABIDE was proposed in 2012 [35] while the state-of-the-art functional preprocessing pipeline is fMRIPrep [36] which introduces less uncontrolled spatial smoothness [36] compared to other preprocessing tools. Some work converts preprocessed neuroimages to brain network datasets which, however, are predominately binarized, i.e., the edge weight can take only two values 0 or 1. For example, in Lanciano et al. [2], the ABIDE dataset was converted to binarized brain networks. In Morris et al. [37], around 80 samples for Attention Deficit Hyperactivity Disorder (ADHD) were released in three datasets KKI, OHSU, and Peking_1 in the form of binarized brain networks. *There still lacks a large collection of quality brain network datasets available to the public.* Our collection uses fMRIPrep on data from 6 sources (see Chapter 4 for details) on 4 clinical conditions of interest under different parcellation schemes and wraps the whole conversion process from raw MRI images to brain

networks in a holistic manner. We release the codes of the entire processing pipeline and continue future efforts to refine the pipeline and/or enrich the collection.

2.1.2 Benchmark.

Network neuroscience that uses machine learning and graph analytics has attracted increasing attention [38]. Along this line of research, most recent studies [2, 39–42] apply machine learning models to perform connectivity analysis on the ABIDE dataset with the generated brain networks. In our benchmark, we tested our datasets on one recent graph analysis model [2], 6 conventional ML models and 6 representative graph ML models: the quality of our datasets is not compromised by the conversion process and can serve as baselines for this line of research.

2.2 General-purposed GNNs

2.2.1 GNNs with Attention Mechanism

The integration of attention mechanisms in GNNs [43, 44] involves incorporating attention-based mechanisms to selectively weigh the importance of different neighbors during the information aggregation process. This attention mechanism enhances the modeling of relationships between nodes in a graph, allowing the model to focus on more relevant neighbors. The relevancy of a neighbor is decided by the dynamic and automated assignment of the attention weights. Such an assignment is based on the content and features of neighboring nodes, which enables the GNN to capture complex and non-linear dependencies in graph-structured data, and thus leads to enhanced performances in various tasks [45, 46]. GAT [43] first introduces the attention mechanism to GNN to utilize the similarity of two nodes to control the weight of the edge for message passing. GAM [47] proposes structural attention to enhance graph structure learning for graph classification. MAGNA [48] proposes a way to incorporate multi-hop context information into every layer of attention computation. CAT [49] incorporates various structural interventions, such as node cluster embedding, and higher-order structural correlations that can be learned outside of GNN, when computing attention scores. GAMLP [50] captures the underlying correlations between different scales of graph knowledge. GraphHAM [51] performs group-level and individual-level attentions when aggregating neighboring states to generate node embeddings.

2.2.2 Graph Contrastive Learning

Recent breakthroughs in contrastive learning, such as Deep InfoMax [52], MoCo [53], and SimCLR [54], have highlighted the potential of discriminative models for representation learning. Contrastive learning focuses on the principle of "learning to compare" by leveraging an InfoNCE objective, which can be formulated as:

$$\mathcal{L} = \mathbb{E}_{x,x^+,x^k}\left[-\log\left(\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + \sum_{k=1}^{K} e^{f(x)^T f(x^k)}}\right)\right] \tag{2.1}$$

In this framework, x^+ is a sample similar to x, x^- is a sample dissimilar to x, and $f(\cdot)$ is an encoder. The similarity measure and encoder may vary depending on the task, but the overarching framework remains consistent across different applications.

In graph contrastive learning, positive and negative samples are generated by data augmentation at the graph structure or node feature level [55]. SUGAR [56] generates subgraphs and uses these subgraphs for reconstruction. AutoGCL [57] employs a set of learnable graph view generators orchestrated by an auto augmentation strategy, where every graph view generator learns a probability distribution of graphs conditioned by the input. ASP [58] proposes a novel attribute and structure preserving graph contrastive learning framework. The objective of these methods is to maximize the similarities of positive pairs and minimize the similarities of negative pairs. However, the relationship between classes is not well-considered, and the data augmentation methods designed for general graph-structured data are hard to apply to brain networks.

2.2.3 Graph Pooling

Graph pooling is a technique in graph neural networks that downsamples graphs by aggregating information from groups of nodes, thereby reducing graph size while preserving essential structural and contextual features for efficient and effective learning. Existing pooling methods can be categorized into node drop pooling and node clustering pooling [59].

Node drop pooling employs learnable scoring functions to eliminate nodes with comparatively lower significance scores. This process can be described as a framework consisting of three modules: (1) Score Generator: This module calculates significance scores for each node

in a given input graph. (2) Node Selector: This module selects the nodes with the top-k significance scores. (3) Graph Coarsening: Using the selected nodes, this module creates a coarsened graph by learning a new feature matrix and an adjacency matrix. Intuitively, methods tend to design more sophisticated score generators and more reasonable graph coarsening techniques to select more representative nodes and retain important structural information, thus mitigating the problem of information loss. Unlike TopKPool [60], SAGPool [61], and HGPSLPool [62], which predict scores from a single view, GSAPool [63] and TAPool [64] generate scores from two different views: local and global. Most methods simply adopt top-k as a selector, with only a few works [65, 66] designing different selectors. Instead of directly obtaining the coarsened graph from the selected nodes as in TopKPool [60], SAGPool [61], and TAPool [64], methods like RepPool [65], GSAPool [63], and IPool [67] utilize both the selected and non-selected nodes to maintain more structural and feature information in the graph.

Node clustering pooling treats graph pooling as a node clustering problem, where nodes are grouped into clusters that are then treated as new nodes in a coarsened graph. This process can be described in two main modules: (1) Cluster Assignment Matrix (CAM) Generator: Given an input graph, the CAM generator predicts the soft or hard assignment for each node. (2) Graph Coarsening: Using the assignment matrix, a new graph is coarsened from the original one by learning a new feature matrix and adjacency matrix. Most existing node clustering pooling methods, which use the same coarsening module, mainly differ in how the CAM is generated. For example, DiffPool [10] directly employs GNN models; StructPool [68] extends DiffPool by explicitly capturing high-order structural relationships; LaPool [69] and MinCutPool [70] design the generator from the perspective of spectral clustering; MemPool [71] introduces a clustering-friendly distribution to generate the cluster matrix.

However, these methods consider the feature and structural information of each graph individually without contrasting the similarity or discrepancy across groups.

2.2.4 Graph Transformer

An alternative method for graph representation learning involves Transformer based models [72], which adapt the attention mechanism to consider global information for each node and incorporate positional encoding to capture graph topological information. Graph Transformers

have garnered significant attention due to their impressive performance in graph representation learning. Dwivedi et al. [73] introduced edge information into the attention mechanism and used eigenvectors as positional embeddings. SAN [74] implemented an invariant aggregation of Laplacian's eigenvectors for positional embedding and introduced conditional attention for real and virtual edges within a graph. Graphormer [75] enhanced the attention mechanism with centrality-based positional embedding and introduced pair-wise graph distances to define relative positional encodings. More recently, GPS [76] proposed a hybrid architecture that combines GNN and Transformer components, achieving state-of-the-art results on various datasets by introducing different types of global/local/relative positional/structural embeddings. Nonetheless, applying these Transformer-based models to brain networks presents challenges [77], primarily due to the correlation-based edges that hinder the use of designs like centrality [78], spatial [75], and edge encoding [76].

2.3 Models for Brain Networks

2.3.1 Single-atlas Methods

In recent years, several GNN-based methods have been proposed for brain networks with single atlas. Ktena et al. [23] leverages graph convolutional networks (GCNs) for learning similarities between each pair of graphs (subjects). BrainNetCNN [20] proposes edge-to-edge, edge-to-node and node-to-graph convolutional filters to leverage the topological information of brain networks in the neural network. LiNet [79] puts forward a two-stage pipeline to discover ASD brain biomarkers from task-fMRI using GNNs. BrainGNN [25] proposes an ROI-selection pooling to highlight salient ROIs for each individual. MG2G [80] is a two-stage approach. The first stage learns node representations through a self-supervised link prediction task. The second stage employs the learned representations to train a classifier for predicting Alzheimer's disease progression. These works neglect the three characteristics of fMRI data elaborated in Chapter 1. Besides, PRGNN [24] proposes a graph pooling method with group-level regularization to guarantee group-level consistency. GroupINN [26] jointly learns the node grouping and extracts the graph features. These two methods only take group information into account on graph level without utilizing node alignment. Lanciano et al. [2] propose a feature extraction method to extract a dense contrast subgraph and filter useful information for prediction. However, their feature extraction and subject classification treat all ROIs and all subjects equally,

which could be vulnerable to noisy data. LG-GNN [81] incorporates local ROI-GNN and global subject-GNN guided by non-imaging data, such as gender, age, and acquisition site. The local ROI-GNN does not take the node alignment of brain networks into account. Moreover, STAGIN [82] and TBDS [83] propose a graph generator to transform the raw BOLD signals into task-aware brain connectivities. They model subjects as dynamic graphs, which require all subjects to have the same scan length and preferably from the same site. In contrast, our work constructs static brain networks as we operate on datasets collected from multi-sites with various acquisition lengths. A Transformer-based method [77] has been applied to brain networks to learn pairwise connection strengths among ROIs across individuals. It neglects the group information of subjects in its methodology design.

2.3.2 Multi-atlas Methods

Multi-atlas methods introduce multiple brain atlases for each neuroimage, which can provide information that complements each other and offers ample details without being restricted by the parcellation mode. MGRL [31] pioneers the construction of multi-atlas brain networks using various atlases. It applies GCNs to learn multi-atlas representations and perform graph-level fusion for disease classification. METAFormer [32] proposes a multi-atlas enhanced transformer approach with self-supervised pre-training for ASD classification. A graph-level late fusion is utilized to aggregate the representations of different atlases. Lee et al. [33] employs a multi-atlas fusion approach that integrates early fusion on the raw feature to capture complex brain network patterns. STW-MHGCN [84] constructs a spatial and temporal weighted hyper-connectivity network to fuse multi-atlas information, and Huang et al. [85] adopt a voting strategy to integrate the classification results of different classifiers (each corresponding to a different atlas) for ASD diagnosis. BrainGT [86] assigns ROIs in different atlases to specific functional regions using coordinates and applies an attention mechanism to learn important features within the same functional region. CP-GCN [87] obtains multi-atlas embeddings by concatenating both intra- and inter-atlas embeddings, followed by a multi-head similarity learning module to construct a population graph for node classification. RAFFNet [88] assigns ROIs from different atlases to predefined functional regions and computes local and global attention to facilitate multi-level information exchange. MHGEL [89]

and CcSiMHAHGEL [90] encode brain networks as hypergraphs to capture higher-order relationships among multiple ROIs and apply consistency regularization to ensure class consistency. MADE-for-ASD [91] uses a bagging ensemble approach to form the final prediction through weighted ensemble voting of multi-atlas representations. However, these studies have not considered the inherent consistency between atlases. Independently encoding multi-atlas brain networks without constraints might extract atlas-specific information, distracting from disease-related pattern modeling. Moreover, existing works only incorporate primitive early or late feature fusion between atlases. This absence of intermediate ROI-level interaction could hinder their models' ability to discern complementary information in each atlas.

2.3.3 Multi-modal and Multi-resolution Methods

Multi-modal and multi-resolution methods explore brain networks using various atlases. Research about multi-modal brain networks [92–101] employs multiple modalities of neuroimaging data, including fMRI, Diffusion Tensor Imaging (DTI) and Positron Emission Tomography (PET), with various atlases to enhance brain network classification, as different modalities provide abundant information compared to a single modality. However, these multi-modal methods focus on fusing structural and functional connectivity information instead of trying to capture the whole picture of the single modality data. Another line of research [102–104] focuses on applying multi-resolution atlases to fMRI data to capture individual behavior across coarse-to-fine scales. However, the technical design of these approaches focuses on extracting information from both fine and coarse scales under the same parcellation mode. Although multi-modal and multi-resolution methods employ various atlases, they focus on different objectives from multi-atlas approaches, and the field of brain network analysis with multi-atlas is still in its infancy stage.

2.4 Summary

In this chapter, we reviewed the literature related to brain network analysis and graph representation learning. We introduced the data collection and benchmark for the brain networks. We then discussed the literature on general-purposed GNNs and models for brain networks. Overall, this chapter provided the background on brain networks needed to understand the data and

network architectures proposed in later chapters of this thesis, as well as some of the challenges that this thesis aims to address.

Chapter 3

Preliminaries

In this chapter, we first formally define the brain network classification task that we focus on in this thesis. We then introduce the commonly-used GNN scheme and Transformer model for graph classification. Notation-wise, we use calligraphic letters (e.g., $\mathcal{G}$) to denote sets, bold capital letters (e.g., $\boldsymbol{A}$) matrices, and bold lowercase letters (e.g., $\boldsymbol{z}$) vectors. Subscripts and superscripts are used to distinguish between different variables or parameters, while lowercase letters denote scalars. We use $\boldsymbol{A}(i,:)$ and $\boldsymbol{A}(:,j)$ to denote the i-th row and j-th column of a matrix $\boldsymbol{A}$, respectively. This notation also extends to a 3D matrix.

3.1 Problem Definition

We represent a graph as $G = (\boldsymbol{A}, \boldsymbol{X})$, where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is its adjacency matrix, and $\boldsymbol{X} \in \mathbb{R}^{n\times c}$ denotes the feature matrix with each node characterized by a feature vector of c dimensions. The node set of G is denoted by $\mathcal{V}_G$ and $|\mathcal{V}_G| = n$. We use $\boldsymbol{X}_v$ to denote the feature vector of a node $v \in \mathcal{V}_G$.

Given a data set of labeled graphs $\mathcal{D} = (\mathcal{G}, \mathcal{Y}) = \{(G, y_G)\}$, where $y_G \in \mathcal{Y}$ is the corresponding label of graph $G \in \mathcal{G}$, the problem of graph classification aims to learn a predictive function $f\colon \mathcal{G} \to \mathcal{Y}$ that maps graphs to their labels.

3.2 Message-Passing Neural Network

Compared with conventional vector-based machine learning models, GNNs engage graph topological information in graph representation learning. The l-th layer of a GNN in the message-

Fig. 3.1: Message-passing neural networks.

passing scheme [105] can be written as:

$$\boldsymbol{H}_v^{(l)} = \mathrm{AGG}^{(l-1)}\left(\boldsymbol{H}_v^{(l-1)}, \mathrm{MSG}^{(l-1)}\left(\left\{\boldsymbol{H}_u^{(l-1)}\right\}_{\forall u \in \mathcal{N}(v)}\right)\right). \tag{3.1}$$

Herein, $\boldsymbol{H}^{(l)} \in \mathbb{R}^{n \times d}$ denotes the l-th layer node representation, where each node is represented by a d dimensional vector. AGG$(\cdot)$ and MSG$(\cdot)$ are arbitrary differentiable aggregate and message functions (e.g., a multilayer perceptron (MLP) can be used as AGG$(\cdot)$ and a summation function as MSG$(\cdot)$). $\mathcal{N}(v)$ represents the neighbor node set of node $v \in \mathcal{V}_G$, and $\boldsymbol{H}_v^{(0)} = \boldsymbol{X}_v$. The updated representations are then passed through a sum/mean pooling and fed to a linear layer for classification. Fig. 3.1 illustrates the message-passing scheme.[1] Note that the update equation is local—it depends solely on the neighborhood $\mathcal{N}(v)$ of node i and is independent of the overall graph size—resulting in a space/time complexity of $O(E)$ that reduces to $O(n)$ for sparse graphs. As a result, MPNNs are highly parallelizable on GPUs and are efficiently implemented using sparse matrix multiplications in modern graph machine learning frameworks [106, 107]. This formulation of MPNNs, also known as Graph Convolutional Networks (GCNs), draws parallels to convolutional neural networks (CNNs) in computer vision [108] by applying a convolution operation with shared weights across the graph domain. In brain network classification, we use the connectivity matrix $\boldsymbol{M}$ as both the adjacency matrix $\boldsymbol{A}$ and feature matrix $\boldsymbol{X}$.

3.3 Transformer

The Transformer architecture consists of a composition of Transformer layers [72]. Fig. 3.2 illustrates the architecture of a single layer of Transformer. Each Transformer layer has two

[1] Figure adapted from https://www.v7labs.com/blog/graph-neural-networks-guide

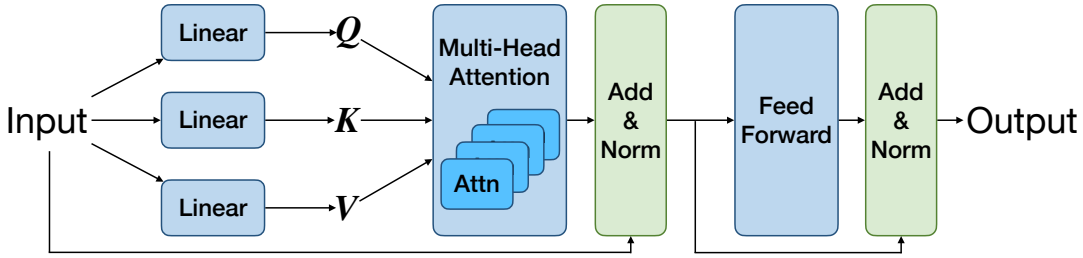


Fig. 3.2: The architecture of a single layer of Transformer.

parts: a self-attention module and a position-wise feed-forward network (FFN). Let $\boldsymbol{H} = \left[\boldsymbol{h}_1^\top, \cdots, \boldsymbol{h}_n^\top\right]^\top \in \mathbb{R}^{n\times d}$ denote the input of self-attention module and $\boldsymbol{h}_i \in \mathbb{R}^{1\times d}$ is the hidden representation at position i. The input $\boldsymbol{H}$ is projected by three matrices $\boldsymbol{W}_Q \in \mathbb{R}^{d\times d_K}$, $\boldsymbol{W}_K \in \mathbb{R}^{d\times d_K}$ and $\boldsymbol{W}_V \in \mathbb{R}^{d\times d_V}$ to the corresponding representations $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$. The self-attention is then calculated as:

$$\boldsymbol{Q} = \boldsymbol{H}\boldsymbol{W}_Q, \quad \boldsymbol{K} = \boldsymbol{H}\boldsymbol{W}_K, \quad \boldsymbol{V} = \boldsymbol{H}\boldsymbol{W}_V, \tag{3.2}$$

$$\text{Attn}\left(\boldsymbol{H}\right) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_K}}\right)\boldsymbol{V}. \tag{3.3}$$

For simplicity of illustration, we consider the single-head self-attention and assume $d_K = d_V = d$. The extension to the multi-head attention is standard and straightforward, and we omit bias terms for simplicity.

Part I

Dataset Construction and Benchmark for Brain Networks

Chapter 4

Dataset and Benchmark

Network neuroscience leverages graph-based machine learning for clinically important applications. For example, a community in brain networks could represent areas of co-activation that could be weakened on brains with neurodegenerative conditions [2]. Graph classification [21] can help differentiate subjects with neurological diseases from healthy ones. Graph ordinal regression [22] can identify subjects with different stages of neurological diseases based on their severity. Nonetheless, *the potential of graph-based machine learning in clinical applications is hindered by the scarcity of available brain network datasets in this important interdisciplinary field.*

In this chapter[1], we focus on fMRI, which monitors changes in the blood flow, i.e., BOLD signals to capture functional activities [4]. The conversion of fMRI scans to brain networks has two stages, preprocessing and brain network construction, both requiring intensive domain inputs. Specifically, to ensure the image quality for subsequent tasks, the preprocessing of raw MRI images needs proper quality control over a number of steps, including motion correction, realigning, field unwarping, normalization, bias field correction, and brain extraction. Different preprocessing choices lead to a large variation in the output images. Parcellation translates the preprocessed MRI images to ROIs as nodes and the co-activation between ROIs as weighted edges. Each generated brain network contains i) a weighted adjacency matrix that characterizes the connectivity between ROIs and ii) a feature matrix that captures the attributes of ROIs in terms of the aggregated BOLD signals. Choosing a different brain atlas/scheme for parcellation

[1]The work in this chapter has been published as "Data-driven network neuroscience: On data collection and benchmark", in Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2023. [6]

leads to a different brain network. As shown in Table 4.1, the existing works of brain network analysis use various datasets, some of which are private, such as Biopoint, PNC and NMU. The parcellation methods they used for different datasets are also inconsistent. Existing study typically selects a single parcellation scheme to generate a fixed set of nodes for all subjects in the same dataset, while the effect of different choices of group-wise data-driven parcellation schemes remains largely unexplored.

Table 4.1: The datasets and parcellation methods used in some of the existing works about brain network analysis.

Method	Dataset	Parcellation/#ROI
BrainGNN [25]	Biopoint [109], HCP [110]	Desikan-Killiany84 [111], Greene268 [112]
PRGNN [24]	Biopoint [109]	Desikan-Killiany84 [111]
LiNet [79]	Biopoint [109]	Destrieux74 [113], Desikan-Killiany84 [111]
EGAT [92]	Biopoint [109]	Lausanne129 [114]
Metric-GCN [23]	ABIDE [34]	HO110 [28]
BrainNetCNN [20]	UBC [115]	UNC90 [116]
FBNetGen [117]	PNC [118], ABCD [119]	Power264 [120], HCP360 [121]
LG-GNN [81]	ABIDE [34], ADNI [122]	HO [28]
BrainTGL [123]	ABIDE [34], HCP [110], NMU	CC200 [124]
BNTF [77]	ABIDE [34], ABCD [119]	CC200 [124], HCP360 [121]
BrainGB [125]	HIV [126], PNC [118], PPMI [127], ABCD [119]	AAL116 [27], Power264 [120], Desikan-Killiany84 [111], HCP360 [121]

The above conversion poses a high barrier to entry the research on brain networks, limiting the development of data-driven network neuroscience. Specifically, domain knowledge in neuroimage preprocessing is required to select and guide the proper pipeline and tools used, image processing and graph extraction lead to high computational costs, and large-scale imaging studies require a complex setup with multi-site, multi-scanner, and multiple acquisition protocols. This chapter aims to bridge the gap by making more brain network data available to the public. We believe that releasing this brain network collection will promote research in the interdisciplinary field of network neuroscience, machine learning, and graph analytics, and advance graph-based and clinical studies such as the detection of neurodegenerative conditions.

4.1 Dataset Sources: Raw Neuroimages

This section describes our selected sources of raw neuroimages and their selection and acquisition settings. Table 4.2 summarizes our datasets and the generated brain networks. There

Table 4.2: Statistics of our datasets and the generated resting-state functional brain networks. Each subject has a graph (brain connectivity network) generated under each Parcellation Method (PM) of AAL, HarvardOxford (HO), Schaefer, k-means and Ward Clustering (see Table 4.4 for details). The number of nodes in a graph generated under a PM is the number of ROIs of the PM. We call an edge non-zero if its weight has absolute value $> 10^{-2}$. The number of non-zero edges varies under different parcellations. The number of node features is the length of the BOLD signals.

Dataset	**Condition**	**# of Graphs (# of Subjects)**	**# of Classes**	**Avg # Non-Zero Edges under PM (# of Nodes)**					**Avg # of Node Features**
				AAL (116)	HO (48)	Schaefer (100)	*k*-means (100)	Ward (100)	
ABIDE	Autism	1025	2	6402	1112	4811	4698	4729	201
ADNI	Alzheimer	1327	6	6447	1112	4824	4734	4715	344
PPMI	Parkinson	209	4	6512	1122	4866	4795	4684	198
Mātai	mTBI	60	2	6433	1112	4832	4750	4731	198
TaoWu	Parkinson	40	2	6481	1116	4846	4724	4766	239
Neurocon	Parkinson	41	2	6455	1114	4830	4677	4779	137

is a class label associated with each subject. The class distribution of each dataset is shown in Table 4.3. This thesis focuses on resting-state functional connectivity of the brain using rs-fMRI, a potent method for detecting neurodegenerative conditions [4], leaving alternative modalities for future exploration. To achieve quality image preprocessing, each rs-fMRI image needs a structural T1-weighted (T1w) image that was acquired from the same subject in the same scan session. T1w image provides structural details which allow brain mask extraction, image alignment and BOLD time series normalization. Other pipelines such as DPARSF [128] have the same requirement.

Autism Brain Imaging Data Exchange (ABIDE) The ABIDE initiative aggregated functional brain imaging data collected from laboratories around the world to support the research on Autism Spectrum Disorder (ASD). ASD has stereotyped behaviors such as irritability, hyperactivity, depression, and anxiety. Subjects are classified into typical controls and those suffering from ASD.

Alzheimer's Disease Neuroimaging Initiative (ADNI) ADNI [129–131] is a longitudinal multisite study for the early detection and tracking of Alzheimer's Disease (AD). AD is a progressive neurologic disorder that causes the brain to shrink and brain cells to die and is the most common cause of dementia that affects a person's ability to function independently. ADNI data used in this study were obtained from the ADNI database (adni.loni.usc.edu) which was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W.

Table 4.3: Our Collection of Brain Network Datasets: Class Distribution

Dataset	Gender (F/M)	Age (mean ± std)	Class	# Subjects
ABIDE	152/873	16.5 ± 7.4	Control	537
			ASD	488
ADNI	728/599	74.6 ± 7.9	CN	819
			SMC	73
			LMCI	102
			EMCI	89
			MCI	179
			AD	65
PPMI	82/127	62.9 ± 9.5	Control	15
			SWEDD	14
			Prodromal	67
			PD	113
Mātai	N/A	N/A	Pre-season	35
			Post-season	25
TaoWu	17/23	65.0 ± 5.0	Control	20
			PD	20
Neurocon	22/19	68.0 ± 11.0	Control	15
			PD	26

Weiner, MD. The primary goal of ADNI has been to test whether serial MRI, PET, other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). Subjects are from 6 different stages of AD: cognitive normal (CN), significant memory concern (SMC), mild cognitive impairment (MCI), early MCI (EMCI), late MCI (LMCI), and Alzheimer's disease (AD) [132].

Parkinson's Progression Markers Initiative (PPMI) PPMI aims to identify biological markers of Parkinson's risk, onset and progression. Parkinson's disease is a progressive nervous system disorder that mainly affects movement [133]. The study is ongoing and contains multimodal, multi-site MRI images similar to ADNI. The PPMI dataset contains subjects from 4 classes: normal control, scans without evidence of dopaminergic deficit (SWEDD), prodromal, and Parkinson's disease (PD).

Mātai Mātai is a longitudinal single site, single scanner study designed for detecting subtle changes in the brain due to a season of playing contact sports. This new dataset consists of the brain networks preprocessed from the data collected from Gisborne-Tairāwhiti area,

New Zealand, with 35 contact sport players imaged at pre-season (N=35) and post-season (N=25) with subtle brain changes confirmed using diffusion imaging study due to playing contact sports. Note that this dataset does not release raw data nor metadata and has been preprocessed so that no ancestral history is able to be extracted. We acknowledge that all data has an origin of significance (Whakapapa).

TaoWu and Neurocon TaoWu and Neurocon datasets are released by ICI [127] and are two of the earliest image datasets released for Parkinson's. The datasets consist of age-matched subjects captured using a single machine and on a single site. We include these two datasets in our collection as they could be used in studies that aim to minimize or contrast the variability introduced from different image acquisition settings. It includes normal controls and patients labelled with PD. Neurocon and Taowu label patients with a diagnosis of PD who have been under treatment (most under levodopa) as PD. PPMI's PD definition involves patients with a diagnosis of PD for two years or less and who are not taking PD medications. Under these definitions, Neurocon and Taowu are more similar when compared to PPMI. It is worth noting that while these two have similar scanning protocols, they used different scanners (with Taowu being higher in resolution). In [127], the authors compared these scans and argued that they can be treated similarly. We believe there could be more such explorations with the data available, which is one of the main reasons why we want to release this collection.

4.2 From MRI Images to Brain Networks: Design Choices

We adopt the common functional processing pipeline in network neuroscience [134] to convert raw MRI images (rs-fMRI and T1w) into brain networks, as depicted in Fig. 4.1. Under this general pipeline, a number of choices need to be made in data selection, data formats, neuroimage preprocessing tools, parcellation schemes, network edge formation, etc. We worked closely with our domain experts to ensure that our design choices are sensible and state-of-the-art. Specifically, Step A collects raw MRI images based on the selection criteria (Section 4.2.1). Step B converts the images into BIDS format (Section 4.2.2). Step C preprocesses the images using fMRIPrep (Section 4.2.3). Step D parcellates the preprocessed data into different ROIs (Section 4.2.4). Steps E-F extract the connectivity matrix and the feature matrix and form the brain network (Section 4.2.5). Domain experts have guided our preprocessing by providing

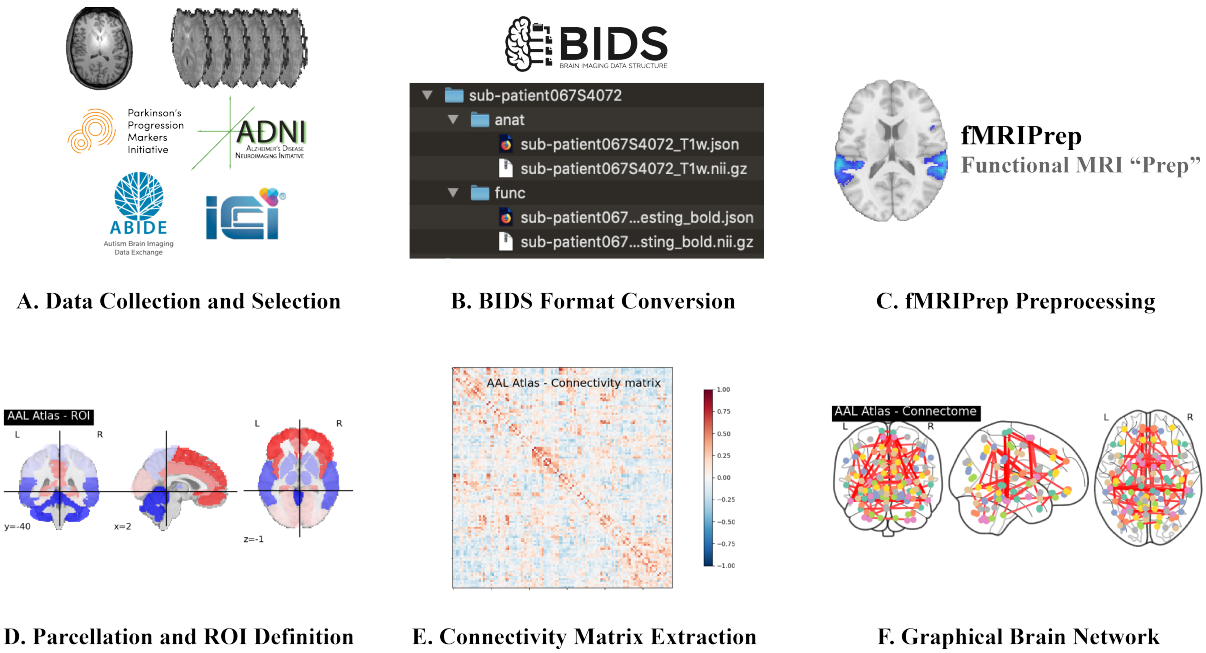


Fig. 4.1: Brain Network Construction Pipeline

advice/feedback on i) the selection of images from the data sources, ii) the choices of using the state-of-the-art fMRIPrep pipeline, iii) parcellation strategies, iv) quality check of fMRIPrep outputs, and v) the selection of confounds.

4.2.1 Data Collection and Selection Criteria

Our data sources involve multi-site, muti-scanner images. Thus, the inclusion criteria of our data collection is based on the availability, validity, and quality of the MRI images of our required types.

For ABIDE, TaoWu, and Neurocon, the original source images were already collated with one raw rs-fMRI image paired with one T1w image for each subject. We included all subjects provided by these data sources except for those that had quality issues. A subject has quality issues if it has an incomplete image (*i.e.*, not containing the full brain) and/or damaged data (*i.e.*, with error reported by any subsequent preprocessing steps). Available metadata of each subject from the source is included in our collection, such as age and gender. The gender distributions of our datasets closely match those of the original data sources. The only exception is ADNI: our released data contains 6% more females than the original ADNI data.

For longitudinal study data sources ADNI, PPMI and Mātai, it is possible that some subjects have had multiple scans over the course of several years. Depending on the scanning protocol for the study, different types of images were taken at various times (*e.g.,* baseline, 1 year follow-up, 2 year follow-up, etc.). The baseline study (the first scan) is usually the most comprehensive one that would cover a wide range of modalities. Thus, as suggested by our domain experts, we consider the baseline study which is likely to be the set of scans with both an rs-fMRI image and a T1w image taken on the same date. In the case that multiple rs-fMRI and T1w scans exist, we selected the first available one.

ABIDE, ADNI, and PPMI are multi-site neuroimaging sources. We chose not to apply data harmonization techniques to these datasets in our preprocessing due to the following reasons. (1) For fMRI connectivity, there is no well-grounded harmonization method for benchmark. (2) Even though no fMRI connectivity harmonization was applied, [135] has shown that fMRI connectivity still has some good repeatability. (3) The site and scanner information are available and researchers have the flexibility of performing harmonization based on our data collection for further analysis.

For validity, we examined each image manually to make sure the images conformed to their labels in the database and did not have obvious data format issues for preprocessing. We had to check this manually because the image databases often had inconsistent labeling.

4.2.2 BIDS Conversion

Raw MRI images are typically in the Digital Imaging and Communications in Medicine (DICOM) format. DICOM images are then converted to the Neuroimaging Informatics Technology Initiative (NIfTI) format using dcm2niix [136], which includes a JSON file that details various imaging parameters such as scanner model, magnetic field strength, flip angle, slice timing, echo time and repetition time. The NIfTI and JSON files are then organized into a folder hierarchy with a precise naming convention known as Brain Imaging Data Structure (BIDS) [137]. After conversion, BIDS formatted data can be applied to preprocessing pipelines in a highly reproducible and transparent manner.

4.2.3 fMRIPrep Preprocessing

BIDS compliant datasets are then preprocessed using fMRIPrep [36], a state-of-the-art tool for preprocessing fMRI. fMRIPrep performs basic processing steps (coregistration, normalization, noise component extraction, skull stripping, etc.) and uses a combination of tools from well-known software packages, including FMRIB Software Library (FSL) [138], Analysis of Functional NeuroImages (AFNI) [139], Advanced Normalization Tools (ANT) [140] and FreeSurfer [141]. This pipeline was designed to use the best software implementation for each step of preprocessing. fMRIPrep can be installed via container technologies such as docker or singularity. fMRIPrep automates the pipeline by using the scanner outputs from the images (*e.g.*, slice timing correction parameters are taken automatically from the scanner outputs of the original image data). Therefore, the user does not need to specify these parameters manually. fMRIPrep can be configured to include or exclude specific workflow steps, such as ignoring slice timing. We followed the default automatic fMRIPrep workflow with surface reconstruction enabled. No further modifications to the settings were made as the default workflow met our expectations for a functional preprocessing pipeline. As part of our validation process of fMRIPrep, we have passed the preprocessed images over to our MRI imaging experts: they have carefully examined the outputs to ensure the data quality. fMRIPrep outputs conform to the BIDS Derivatives specification for compatibility and include the preprocessed BOLD images and the confounds file, which records fluctuations during MRI data acquisition, also known as nuisance regressors. fMRIPrep is computationally expensive. In our case, one run on one subject of fMRIPrep using 8 threads on an Intel(R) Core(TM) i9-10940X CPU@3.30GHz took on average 4-5 hours.

4.2.4 Parcellation Strategies

Parcellation defines regions in the brain known as ROIs. The parcellation step takes in the two outputs from fMRIPrep for each subject to generate the parcellations: the preprocessed BOLD image and the confounds file. We adopted the standard 9 parameters (9P) confounds setting widely used in functional connectivity studies [142, 143] for denoising: white matter, cerebrospinal fluid, global signal, and the 6 rigid-body motion parameters for rotation and

Table 4.4: Parcellation Methods

Name	# ROIs	Generation Method
AAL [27]	116	Delineated with respect to anatomical landmarks by following the sulci course in the brain.
HarvardOxford (HO) [28]	48	Created by subdividing neocortex by topographic criteria into 48 parcellation units corresponding to the principal cerebral gyri.
Schaefer [29]	100	Using gradient-weighted Markov Random Fields (gwMRF) to automatically group similar fMRI regions.
k-means Clustering [144, 145]	100	Top-down clustering algorithm that partitions voxels into non-overlapping predefined number of regions.
Ward Clustering [145, 146]	100	Bottom-up hierarchical clustering algorithm that agglomerates together voxels progressively into regions.

translations at x, y and z axes. To parcellate the brain, we selected a list (Table 4.4) of Parcellation Methods (PMs) that cover both atlas-based and clustering-based PMs frequently used in the domain.

Brain regions partitioned through predefined regions are atlas-based PMs. We selected the most commonly used atlas-based PMs, AAL, HarvardOxford (HO), and Schaefer, that are generated using different strategies and brain features.

Clustering-based parcellations partition the brain by computing the similarity in the BOLD signals between voxels and performing clustering on the voxels based on the similarities. Each subject directly generates its own set of ROIs. *k*-means and Ward clustering are the two methods used in the domain. We included them and set the number of ROIs as 100, which is close to the predefined value in atlas-based PMs. Fig. 4.2 shows examples of the ROIs extracted by these methods.

4.2.5 Brain Network Extraction

The BOLD signals from parcellated ROIs are used to compute the connectivity matrix of the brain network.

The matrix captures functional relationships between parcellated regions and reflects their co-activations. Connectivity matrix is extracted using a built-in function of nilearn [145] called

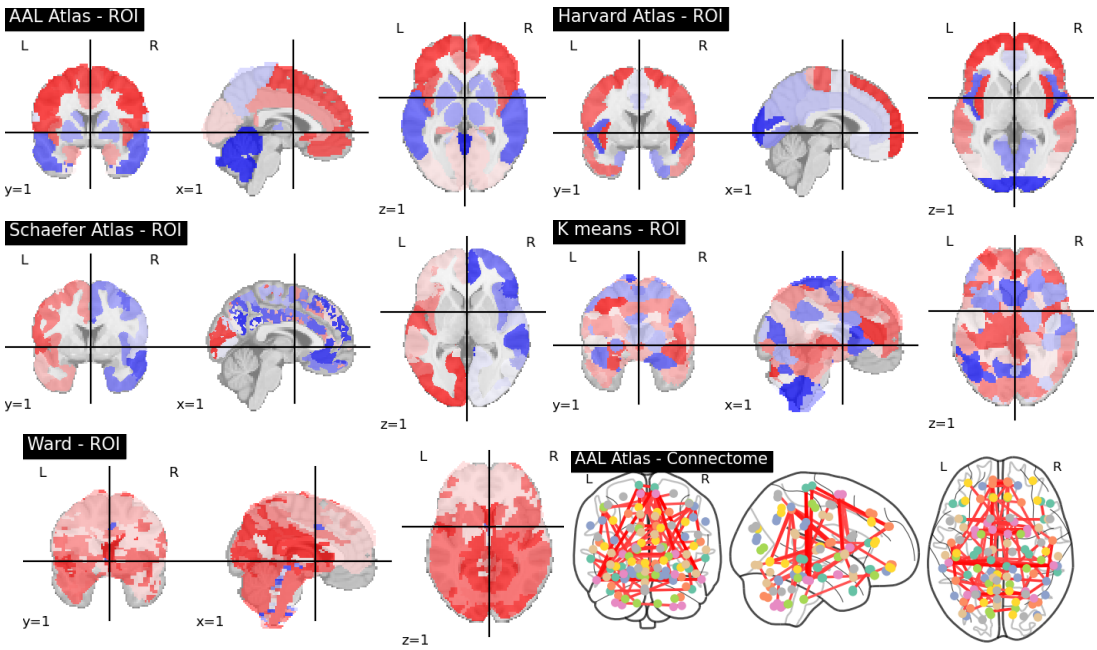


Fig. 4.2: Extracted ROIs and Glass Brain Connectome (right bottom corner)

ConnectivityMeasures [147]. It takes 2 inputs, BOLD signals of the brain regions and the connectivity metric (*e.g.*, correlation). The former is obtained by averaging the BOLD signals of all the voxels lying in the same ROI. Fig. 4.1E shows an example of a connectivity matrix. Among existing connectivity measures (correlation, covariance, partial correlation, etc.), correlation is used by a majority of graph-based neuroscience research [2,41,148]. Thus, in our released collection, we select correlation as the connectivity measure and make the fully weighted matrices available without performing any thresholding on the correlation values. For the practitioners who are interested in alternative measures, we have uploaded preprocessed images (excluding Mātai dataset) and our matrix generation code as part of our collection, which enables the researchers to generate their own matrices based on their desired measure.

The connectivity matrix can be represented equivalently as a glass brain connectome. An example connectome plot on the AAL atlas is shown in the right bottom corner of Fig. 4.2. The connectome plot shows top 1% of brain functional connectivities for visualization.

The brain network constructed by this pipeline has the set of ROIs as nodes and their co-activations (measured as Correlation) as edges. Each brain network is represented by a weighted adjacency matrix and a feature matrix. The former is the connectivity matrix extracted and the latter stores the BOLD signals of ROIs. Note that we include the BOLD signals

as node feature matrices in our data release for data completeness: some analytical methods on brain networks may want to utilize such information directly or further extract features from them.

4.3 Data Quality Assessment and Baseline Comparisons

To assess if the graph representation maintains the data quality from neuroimages, we test our brain network data collection on two tasks, graph classification and graph ordinal regression. The aim of graph classification is to distinguish patients from healthy subjects for datasets with two classes (ABIDE, Mātai, TaoWu, and Neurocon); and to distinguish subjects among different disease stages and healthy groups by treating each class independently when it comes to multi-class datasets (ADNI and PPMI). In contrast, graph ordinal regression is only applied to multi-class datasets by differentiating stages based on the disease severity.

For classification, we test on both conventional ML models typically used in neuroscience and representative graph ML models. For conventional models, we follow the common practice in the domain [39, 42] to vectorize the input connectivity matrices by flattening them. For ordinal regression, we test on the classic logistic ordinal regression model. To further validate the quality of our data, we also test on a recent graph analysis model on functional networks for classification [2]. Finally, we also include a sensitivity study on the number of ROIs and the training set size. All these models only utilize the connectivity matrices for assessing the quality of our brain network construction. The data is split to 8:1:1 for training, validation, and testing with 10-fold cross-validation performed.

4.3.1 Results on Classification

We select 6 conventional machine learning models for this study: Logistic Regression (LR), Gaussian Naive Bayes (NB), Support Vector Machine Classifier (SVC), k-Nearest Neighbours (kNN), Random Forest (RF), and Multi-Layer Perceptron (MLP) [149]. We used the implementation from the scikit-learn library [150, 151] with Grid Search for model selection. We also select 6 typical graph-based machine learning models, including GCN [152], GraphSAGE [153], GIN [105], GAT [43], GatedGCN [154], and BrainNetCNN [20]. The implementations mainly follow those in [155]. To compare the effectiveness of brain networks with that

Table 4.5: Classification accuracy (mean±standard deviation) on conventional ML methods. The best result at each parcellation is highlighted in bold. The best result in each dataset is underlined.

	ABIDE					**ADNI**				
	AAL	HO	Schaefer	*k*-means	Ward	AAL	HO	Schaefer	*k*-means	Ward
LR	63.8±3.0	**63.6±4.2**	**64.8±3.7**	48.4±4.4	51.1±5.8	**64.1±1.8**	61.9±2.1	62.0±4.2	58.6±2.6	60.4±0.8
NB	60.4±5.5	59.2±5.4	61.6±3.6	51.6±3.7	**54.2±5.0**	53.6±4.4	52.7±6.7	48.9±3.3	32.5±3.6	55.0±2.6
SVC	**65.7±3.3**	62.9±3.6	64.4±5.1	49.3±4.1	53.2±4.9	63.4±1.9	**66.2±2.9**	61.5±5.0	**61.8±0.3**	**61.8±0.3**
kNN	58.1±5.3	56.1±4.5	59.7±3.5	50.7±2.9	49.3±2.3	60.6±2.1	62.9±3.8	**63.1±2.4**	31.4±22.9	59.5±3.9
RF	62.4±2.8	60.5±3.1	62.6±2.6	49.5±5.4	51.3±2.9	61.9±2.1	61.7±2.8	62.1±1.8	61.5±0.4	61.6±0.5
MLP	54.4±1.2	62.2±4.4	49.6±4.4	**63.6±3.9**	51.7±6.3	62.4±1.8	62.9±2.2	46.7±5.3	61.7±0.2	48.6±5.9
	PPMI					**Mātai**				
	AAL	HO	Schaefer	*k*-means	Ward	AAL	HO	Schaefer	*k*-means	Ward
LR	56.0±7.8	56.0±9.2	56.5±6.8	60.3±7.4	60.3±8.8	66.7±21.1	**58.3±13.4**	60.0±20.0	55.0± 18.3	**58.3±20.1**
NB	58.4±5.2	52.6±8.6	57.5±7.6	58.4±5.7	**62.2±7.4**	**68.3±22.9**	45.0±21.2	56.7±18.6	55.0±15.0	53.3±18.0
SVC	**64.1±5.7**	**63.6±6.4**	**63.2±8.6**	**60.8±7.5**	60.8±8.9	65.0±20.3	**58.3±13.4**	56.7±17.0	58.3±17.1	**58.3±17.1**
kNN	51.2±9.1	55.5±7.3	53.5±7.6	60.3±6.6	60.8±8.9	61.7±16.8	50.0±21.1	58.3±18.7	58.3±17.1	48.3±13.8
RF	61.6±8.8	62.6±9.9	62.6±8.1	58.4±9.2	61.7±7.3	60.0±18.6	53.3±20.0	**63.3±24.5**	48.3±17.4	53.3±20.8
MLP	57.8±10.4	62.2±8.0	57.9±5.0	57.4±9.2	52.6±5.7	45.0±19.8	48.3±21.7	53.3±18.0	**63.3±18.0**	50.0± 22.4
	TaoWu					**Neurocon**				
	AAL	HO	Schaefer	*k*-means	Ward	AAL	HO	Schaefer	*k*-means	Ward
LR	**77.5±17.5**	**72.5±23.6**	**75.0±15.0**	50.0±15.8	**52.5±17.5**	**68.5±25.1**	61.5±26.3	65.5±24.9	58.0±22.9	63.0±25.9
NB	65.0±15.8	60.0±24.5	62.5±23.6	37.5±30.1	50.0±19.4	58.0±22.9	59.0±28.4	63.5±23.3	48.0±24.2	55.5±22.3
SVC	67.5±17.5	67.5±17.5	65.0±15.0	52.5±7.5	50.0±19.4	63.0±25.9	**68.5±25.9**	**69.0±25.9**	63.0±25.9	63.0±25.9
kNN	55.0±21.8	60.0±16.6	65.0±16.6	42.5±16.0	50.0±0.0	65.0±25.5	49.0±27.6	55.5±24.9	63.0±25.9	63.0±25.9
RF	65.0±11.3	60.0±22.9	57.5±25.1	40.0±32.0	47.5±20.8	55.5±24.9	61.0±29.8	58.5±22.4	58.0±22.9	63.0±25.9
MLP	60.0±20.0	67.5±16.0	42.5±22.5	**57.5±27.5**	45.0±21.8	61.0±11.8	67.5±16.0	63.5±11.8	**63.5±11.8**	**63.5±11.8**

of BOLD signals in disease classification, we test on two methods based on BOLD signals: GRU [156] and 1D-CNN [157].

Table 4.5 reports the classification accuracy of conventional ML methods on the datasets using different parcellation methods. In most cases, the overall accuracy falls within the range of 60% to 70%, which is consistent with the results reported in previous studies [2, 39–42], despite some differences in data filtering and preprocessing. With respect to parcellation methods, we observe that atlas-based parcellation methods in general outperform clustering-based methods. The best performance on each dataset always occurs when AAL, HO or Schaefer is applied. Particularly, AAL achieves the best performance in 4 out of 6 datasets. The inferiority of clustering-based parcellation is likely due to the effects from some randomness in the clustering process, *e.g.*, initial centroid selection in *k*-means. Among different learning models, LR and SVC perform better in most cases. This also explains why they are commonly used as baseline models in the domain [2, 41].

Table 4.6 reports the classification results of graph-base ML methods and BOLD time series

Table 4.6: Classification accuracy (mean±standard deviation) on graph ML methods and BOLD time series based methods with Schaefer parcellation. The best result in each dataset is in bold, with those underlined indicating superior performance to conventional ML methods.

	ABIDE	**ADNI**	**PPMI**	**Mātai**	**TaoWu**	**Neurocon**
GCN	61.0±2.8	61.6±0.6	54.0±9.1	56.7±18.6	60.0±29.2	59.0±20.7
GraphSAGE	63.1±3.1	61.2±1.7	55.0±12.9	61.7±10.7	60.0±33.9	68.5±15.2
GIN	57.0± 3.9	61.9±0.4	**57.9±8.1**	48.3±13.8	65.0±20.0	68.5±15.2
GAT	60.9±5.0	61.3±1.3	55.0±8.0	**66.7±18.3**	**67.5±22.5**	54.0±15.6
GatedGCN	63.6±4.7	**62.1±4.5**	52.6±11.5	58.3±8.3	65.0±22.9	**69.0±25.5**
BrainNetCNN	**65.8±2.5**	61.1±2.9	57.3±10.3	61.7±13.3	65.0±27.8	66.0±22.5
GRU	53.6±1.4	61.8±0.3	54.1±2.2	58.3±8.3	50.0±0.0	63.8±1.5
1D-CNN	55.5±3.0	50.2±4.0	55.0±11.5	61.7±13.0	45.0±15.0	58.5±11.2

based methods on the datasets with the Schaefer parcellation. No single graph ML method can consistently dominate across datasets. In 3 out of 6 datasets, the best graph ML method outperforms the best of conventional ML methods, which shows the potential of graph-based models. Compared with both conventional and graph ML models, the model performance based on BOLD signals is significantly worse. This verifies the effectiveness/usefulness of brain networks in the task of disease classification.

The accuracy scores in Tables 4.5 and 4.6 might appear relatively modest, raising concerns about the clinical applicability of solving the classification problem. It's important to note that disease classification through brain networks is a nascent and evolving research domain. Many of the ML methods tested are general-purpose and not specifically designed for brain networks. Our release of fully weighted brain network matrices in this collection is anticipated to stimulate and advance research in this field. This initiative is expected to foster the development of tailored learning models for brain networks, yielding clinically relevant outcomes. The current results can serve as foundational benchmarks for future research endeavors.

In addition, we provide the results on ordinal regression in Appendix A.1.1. Data quality study on ABIDE dataset with a graph analysis approach for classification is also provided in Appendix A.1.2.

Table 4.7: Classification accuracy (mean±standard deviation) when tuning #ROIs in Schaefer parcellation. The best result in each method is in bold and the best result in each dataset is underlined.

	ABIDE				**TaoWu**			
	100	200	500	1000	100	200	500	1000
LR	64.8±3.7	**67.9±3.8**	67.4 ±3.8	67.6±4.7	**75.0±15.0**	57.5±22.5	52.5±23.6	52.5±17.5
NB	61.6±3.6	**62.4±3.6**	61.9±3.0	60.5±3.5	**62.5±23.6**	**62.5±23.1**	60.0±20.0	60.0±20.0
SVC	**64.4±5.1**	63.7±4.4	62.4±3.8	63.1±4.0	**65.0±15.0**	57.5±22.5	55.0±24.5	50.0±19.4
kNN	**59.7±3.5**	59.5±5.1	58.0±3.8	55.5±5.8	**65.0±16.6**	52.5±23.6	50.0±25.0	50.0±25.0
RF	**62.6±2.6**	62.4±3.2	61.6±3.2	62.1±5.3	**57.5±25.1**	55.0±26.9	52.5±26.1	45.0±18.7

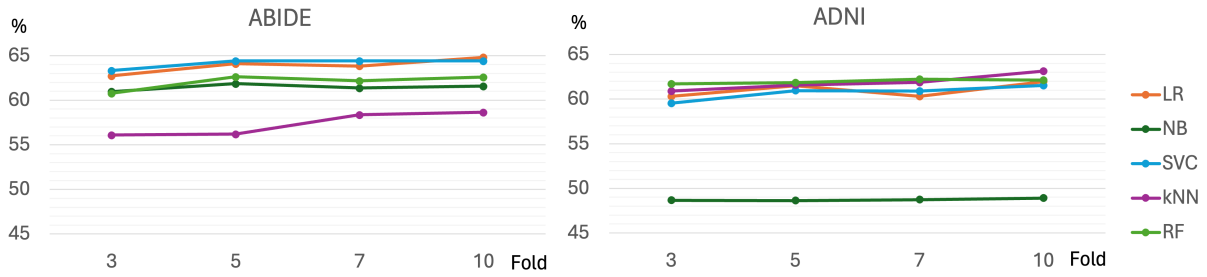


Fig. 4.3: Test accuracy on ABIDE and ADNI with Schaefer when tuning training set size.

4.3.2 Sensitivity Study on Number of ROIs and Training Set Size

To assess the effect of the number of ROIs to model performance, we tune the number of ROIs on the Schaefer parcellation from {100, 200, 500, 1000} on the classification task. The results are reported in Table 4.7. Each model performs the best when the number of ROIs is set at 100 or 200.

We also test on different training set sizes by tuning k in k-fold CV and report the results in Fig. 4.3. The performance of each method improves with the increase in the training set size (larger k), which confirms the necessity of having datasets at this scale.

4.4 Summary

This thesis release a functional brain network collection to the public at a large scale.[2] The collection originates from 6 raw MRI image sources, covers 4 brain conditions, and totals to

[2] https://doi.org/10.17608/k6.auckland.21397377

2,702 subjects. Working with domain experts, this thesis come up with a unified pipeline that converts raw fMRI and T1w images to brain networks. We tested the collection on 12 ML models and a recent graph analysis model to demonstrate that the data quality is not compromised while at the same time providing the results as domain baselines. We hope that the release of this collection of brain networks, together with the complete code of the processing pipeline, will promote both the development of graph-based models and the clinical advancement in the diagnosis and early intervention of neurodegenerative diseases.

Part II

Single-atlas Brain Network Analysis

Chapter 5

Class-Aware Representation Refinement Framework

In this chapter[1], we tackle the problems of (1) neglect of graph-level relationships and (2) generalization issue. Each graph is treated independently in GNN message passing/graph pooling, and existing methods to address overfitting operate on each individual graph. This makes the graph representations learnt less effective in the downstream classification. In this chapter, we propose a Class-Aware Representation rEfinement (CARE) framework for the task of graph classification. CARE computes simple yet powerful class representations and injects them to steer the learning of graph representations toward better class separability. CARE is a plug-and-play framework that is highly flexible and able to incorporate arbitrary GNN backbones without significantly increasing the computational cost. This chapter also theoretically prove that CARE has a better generalization upper bound than its GNN backbone through Vapnik-Chervonenkis (VC) dimension analysis. Our extensive experiments with 11 well-known GNN backbones on 14 benchmark datasets validate the superiority and effectiveness of CARE over its GNN counterparts.

5.1 Drawbacks of Graph Classification

These existing GNNs suffer from two major drawbacks when applied to the downstream classification task: (1) Neglect of graph-level relationships; and (2) Generalization issue.

[1]The work in this chapter has been published as "A Class-Aware Representation Refinement Framework for Graph Classification", Information Sciences, vol. 679, p. 121061, 2024. [158]

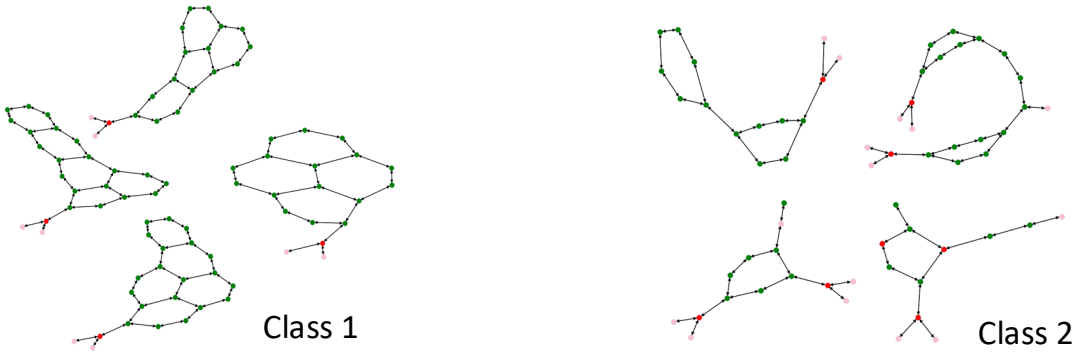


Fig. 5.1: An example of molecular data in different classes from MUTAG dataset.

Neglect of graph-level relationships. Existing GNN architectures consider each input graph independently in their training processes. Input graphs are passed individually to GNN to yield node representations. In addition, the model also treats each graph separately in its design of loss. The relationships (similarity and/or discrepancy) among different input graphs are fully neglected. Though the model parameters are trained by the set of input graphs collectively, it is done through a long pathway from node representations to graph representations and finally to the loss. As a result, the effectiveness of the model will be significantly compromised when applied to the downstream classification. With molecular data, for instance, one would want the molecules from the same class to share similar representations. This is natural as molecules belonging to the same class often carry certain common substructures (e.g., the same set of functional groups). A more specific example of molecular data from MUTAG dataset is provided in Fig. 5.1. Graphs from class 1 all have multiple cycles and they are directly adjacent to each other. But graphs in class two only contain one or two cycles and they are connected by a bridge-like substructure. These graph-level patterns could be class-specific and serve as a perfect signature of a class. Without considering such graph-level information, the graph representations learnt would be less effective in separating different graph classes.

Generalization issue. This is an inherent issue in GNN that the model tends to overfit when the network gets deeper or the hidden dimensionality gets larger [159]. Some methods have been proposed to alleviate this issue. Several graph-argumentation based methods improve the generalization ability by modifying input graphs [160], or generating new graphs for adversarial learning [161] and contrastive learning [55]. Some other works [162, 163] resample or reweight data instances to remit the overfitting problem. However, these methods operate

on each individual graph and fail to explore the effectiveness of graph-level information in improving generalization.

5.2 Methodology

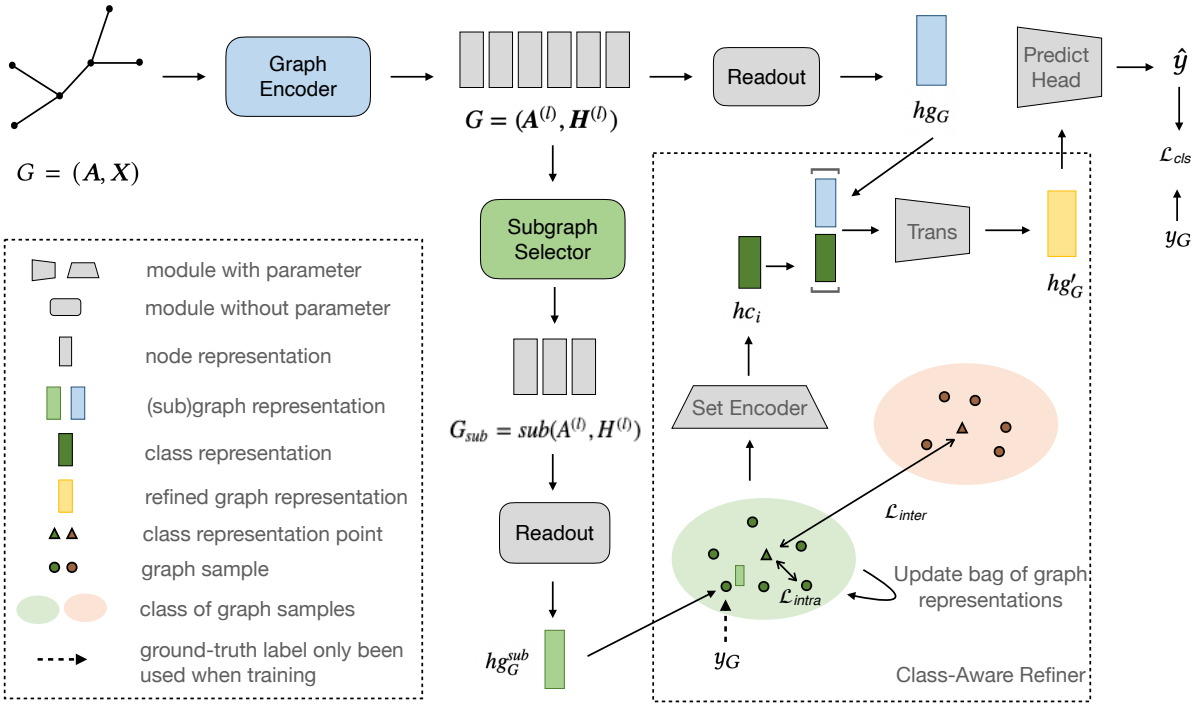


Fig. 5.2: Framework of CARE.

5.2.1 Proposed Framework

We now describe our proposed Class-Aware Representation rEfinement framework (CARE), which aims to refine graph representations by considering the graph-level similarity. CARE contains four main components, including a graph encoder, a subgraph selector, a class-aware refiner and a class loss. The former two allow the flexible incorporation of existing GNN methods, while the latter two are newly proposed in our framework. Fig. 5.2 depicts the CARE framework.

The remainder of this section describes the four components in detail. We first introduce the graph encoder to get the initial node/graph representation, and then describe the subgraph

selector to extract an appropriate substructure for the subsequent class representation learning. The class-aware refiner learns class representations from different graph classes, which are used to refine graph representations. A class loss is proposed to further improve class separability. The two new components in CARE only contain a small number of parameters and are easy to plugin arbitrary GNN backbone.

Graph Encoder. A graph encoder extracts the node representations $\boldsymbol{H}$ and the graph-level representation $\boldsymbol{hg}_G$ for graph G. CARE does not impose any constraint on the architecture of the graph encoder. Any message-passing GNN model could be applied here, which is formalized as

$$\boldsymbol{H}_v^{(l+1)} = \text{UPDATE}^{(l)}\left(\boldsymbol{H}_v^{(l)}, \text{AGG}^{(l)}\left(\left\{\boldsymbol{H}_u^{(l)}\right\}_{\forall u \in \mathcal{N}(v)}\right)\right), \tag{5.1}$$

where $\boldsymbol{H}^{(l+1)} \in \mathbb{R}^{n \times m}$ denotes the $(l+1)$-th layer node representation with m dimensions, UPDATE and AGG are arbitrary differentiable update and aggregate functions, $\mathcal{N}(v)$ represents the neighbor node set of node $v \in \mathcal{V}_G$, and $\boldsymbol{H}_v^{(0)}$ is initialized as the input feature vector $\boldsymbol{X}_v$.

After a few message-passing layers, we can obtain a set of node representations. A READOUT function can be applied to produce the graph representation $\boldsymbol{hg}_G \in \mathbb{R}^m$ as:

$$\boldsymbol{hg}_G = \text{READOUT}\left(\{\boldsymbol{H}_v \mid v \in \mathcal{V}_G\}\right). \tag{5.2}$$

Subgraph Selector. The class-aware refiner in CARE aims to maintain generic features for graph samples from different classes. However, the READOUT function treats all nodes equally without considering the class information. In fact, graphs in different classes are likely to have various substructures. To address this limitation, CARE introduces a subgraph selector sub$(\cdot)$ to filter nodes in the original graph, which is defined as $\boldsymbol{A}^{(l+1)}, \boldsymbol{H}^{(l+1)} = \text{sub}(\boldsymbol{A}^{(l)}, \boldsymbol{H}^{(l)})$.

Any graph pooling methods could be applied here to select subgraphs. Typical ones include node drop pooling methods [61] and node clustering pooling methods [164].

Class-Aware Refiner. As existing GNN models ignore the relationships of graphs from different classes, a new component is designed in CARE to fill this gap. In the training process, the class-aware refiner utilizes the ground truth label of each training instance. It maintains a bag of encoded (sub)graph representations $\mathcal{B}_i$ and aggregates these representations to obtain a class representation $\boldsymbol{hc}_i$ for each class $i \in \mathcal{Y}$. The aggregation function is a universal Set Encoder,

e.g., DeepSets [165] or PointNet [166]. Herein, we apply DeepSets as in Eq. (5.3), in which $\rho(\cdot)$ is a multilayer perceptron (MLP) with a non-linear function ReLU, and $\phi(\boldsymbol{hg}) = \boldsymbol{hg}/|\mathcal{B}_i|$.

$$\boldsymbol{hc}_i = \rho\left(\sum_{\boldsymbol{hg}_G^{sub} \in \mathcal{B}_i} \phi(\boldsymbol{hg}_G^{sub})\right), \tag{5.3}$$

where $\boldsymbol{hg}_G^{sub}$ is the subgraph representation of a graph $G \in \mathcal{G}$, obtained by passing the output of the subgraph selector through the READOUT function. The class representation $\boldsymbol{hc}_i$ is then used to refine graph representations for all graphs in the same class:

$$\boldsymbol{hg}'_G = \text{Trans}([\boldsymbol{hg}_G \mid \boldsymbol{hc}_i]), \tag{5.4}$$

where $\boldsymbol{Trans}(\cdot)$ is a transformation function and $\boldsymbol{hg}'_G$ is the refined graph representation. Herein, we set $\boldsymbol{Trans}(\cdot) = \rho(\cdot)$.

In the validation and the test processes, the ground truth label y_G is not available for a validation/test graph G. However, we need a label for each validation/test graph to determine which class representation to be used to composite its refined representation. In this case, the Class-Aware Refiner will predict a pseudo label $\widetilde{y}_G$ for graph G by classifying it to the most similar class and use the corresponding class representation for graph representation refinement. We use the cosine similarity as a metric to quantify the similarity between a graph representation and a class representation. The pseudo label is obtained as $\widetilde{y}_G = \arg\max_{i \in \mathcal{Y}}(cos\_sim(\boldsymbol{hg}_G^{sub}, \boldsymbol{hc}_i))$. Note that class representations are kept unchanged in the validation/test process. The training algorithm of the class-aware refiner is summarized in Algorithm 1.

Algorithm 1 Training of Class-Aware Refiner

Input: Subraph representation $\boldsymbol{hg}_G^{sub}$ of graph G (output by Subgraph Selector), graph representation $\boldsymbol{hg}_G$, and ground-truth label y_G of G;
Output: Refined graph representation $\boldsymbol{hg}'_G$, intra-class loss $\mathcal{L}_{intra}$ and inter-class loss $\mathcal{L}_{inter}$;

Use $\boldsymbol{hg}_G^{sub}$ to update the bag of (sub)graph representations $\mathcal{B}_i$ for $i = y_G$;
Calculate the class representation $\boldsymbol{hc}_i$ by Eq. (5.3);
Use $\boldsymbol{hc}_i$ and $\boldsymbol{hg}_G$ to obtain the refined graph representation $\boldsymbol{hg}'_G$ by Eq. (5.4);
Calculate the intra-class loss $\mathcal{L}_{intra}$ and inter-class loss $\mathcal{L}_{inter}$ by Eqs. (5.5) and (5.6);

Class Loss. For graph classification task, it would be beneficial to exploit the graph similarity within the same class and the graph discrepancy between different classes. This is essential for

making different classes more separable. Using the classification loss only fails to learn such graph-level relations. Therefore, a class loss $\mathcal{L}_{class}(\cdot)$ is proposed in CARE to enforce the intra-class similarity and the inter-class discrepancy. The former $\mathcal{L}_{intra}$ is defined as the similarity between each graph representation and its class representation, while the latter $\mathcal{L}_{inter}$ is defined as the similarity between different class representations:

$$\mathcal{L}_{intra} = \underset{i \in \mathcal{Y}}{AVG}(\underset{y_G = i}{AVG}(cos\_sim(\boldsymbol{hc}_i, \boldsymbol{hg}_G^{sub}))), \tag{5.5}$$

$$\mathcal{L}_{inter} = \underset{i \in \mathcal{Y}}{AVG}(\underset{j > i, j \in \mathcal{Y}}{AVG}(cos\_sim(\boldsymbol{hc}_i, \boldsymbol{hc}_j))), \tag{5.6}$$

We again use the cosine similarity as a metric. The class loss $\mathcal{L}_{class} = \exp(\mathcal{L}_{inter} - \lambda_1 * \mathcal{L}_{intra})$ is then defined as a function that maximizes $\mathcal{L}_{intra}$ and minimizes $\mathcal{L}_{inter}$, where $AVG(\cdot)$ is the average function and λ_1 is a trade-off hyperparameter.

The predicted class label is still supervised by a classification loss $\mathcal{L}_{cls}(\cdot)$. Herein, we apply the commonly used cross-entropy loss [167]. The overall loss $\mathcal{L}$ of CARE is defined as:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_2 * \mathcal{L}_{class}, \tag{5.7}$$

where λ_2 is a trade-off hyperparameter for balancing the classification loss and the class loss.

5.2.2 Model Architecture Variants

As GNNs can be categorized as hierarchical ones and non-hierarchical ones, we design two corresponding architectures that apply CARE to plug-and-play in different GNN backbones.

Global Architecture. Several GNN models (e.g., GCN [152], GAT [43] and GraphSAGE [153]) apply the readout function only at the end of graph convolution. The global architecture of CARE is designed to apply the Class-Aware Refiner only after the readout function. The outputs are then passed to a linear layer for graph classification.

Hierarchical Architecture. Some other GNN models, such as GIN [105] and UGformer [168], have a readout function in each graph convolutional layer. The graph representations from each layer are taken into account when making the final prediction. The hierarchical architecture of CARE is designed to apply the class-aware refiner on each layer in order to cope with the hierarchical GNN backbones.

5.2.3 Generalization Analysis

In this section, we present the theoretical support that the proposed CARE has a better model generalization than its GNN backbone in the case of binary classification. We use the VC dimension to measure the capacity of a model. Based on the VC theory [169], reducing the VC dimension of a model has the effect of eliminating potential generalization errors.

Our analysis is grounded on the VC theory for neural nets [170]: the VC dimension of a neural network is upper bounded by a function with respect to the number of model parameters t and the number of operations p. In the following, we first derive the computational complexity of the GNN backbone and CARE measured by the number of multiplications, based on which we obtain an upper bound of the VC dimension for each model. We then present a theorem that states that CARE has a lower VC dimension upper bound than its GNN backbone when the number of parameters is identical. In subsequent discussions, we use GCN as an example backbone. The conclusion generally applies to other backbones by plugging in their corresponding computational complexity. We present the theoretical results here and defer the detailed proofs to Appendix B.1.

Computational Complexity of Models. CARE and its backbone GCN are both composed of GCN layers, an embedding layer, and several fully-connected layers.

[*Complexity of GCN Backbone.*] We denote the GCN layer in the GCN model as $\text{gcn}(\cdot)$ and its input/output dimensions as $h_{gcn_{in}}/h_{gcn_{out}}$. The layer mapping is given by $\text{gcn}(\boldsymbol{A}, \boldsymbol{H}) = \sigma_{gcn}(\hat{\boldsymbol{A}}\boldsymbol{H}\boldsymbol{W}_{gcn})$, where σ_{gcn} is the activation function, $\boldsymbol{W}_{gcn} \in \mathbb{R}^{h_{gcn_{in}} \times h_{gcn_{out}}}$ is the weight matrix, and $\hat{\boldsymbol{A}}$ is the normalized adjacency matrix. The computational complexity of the GCN network measured by the multiplication number, denoted as $q_1(d)$, for d number of layers, is given by:

$$q_1(d) = \sum_{l=0}^{d} (n^2 h^l_{gcn_{in}} + n h^l_{gcn_{in}} h^l_{gcn_{out}}). \tag{5.8}$$

[*Complexity of CARE.*] The GCN-based CARE network with a hierarchical architecture is composed of the following:

- a GCN layer same as the GCN backbone, whose computational complexity is $q^l_{gcn} = n^2 h^l_{gcn_{in}} + n h^l_{gcn_{in}} h^l_{gcn_{out}}$.
- a subgraph selector (SAGPool), which contains a scoring layer (GCN with 1-dimensional output) and a top-k pooling algorithm. The complexity is $q^l_{subgraph} = n^2 h^l_{gcn_{out}} + n h^l_{gcn_{out}}$.

- a class-aware refiner contains a set encoder Eq. (5.3) and a transformation layer Eq. (5.4). The mapping of the fully-connection layer fc(·) from input $\boldsymbol{H}$ is given by $\text{fc}(\boldsymbol{H}) = \sigma_{fc}(\boldsymbol{H}\boldsymbol{W}_{fc})$, where σ_{fc} is the activation function, $\boldsymbol{W}_{fc} \in \mathbb{R}^{h_{fc_{in}} \times h_{fc_{out}}}$, and $h_{fc_{in}}/h_{fc_{out}}$ are the input/output dimensions. The complexities for the set encoder and the transformation layer are $q^l_{set} = nh^l_{set_{in}} h^l_{set_{out}}$ and $q^l_{trans} = nh^l_{trans_{in}} h^l_{trans_{out}}$, respectively.

Therefore, the computational complexity of the GCN-based CARE is given by:

$$q_2(d) = \sum_{l=0}^{d} (q^l_{gcn} + q^l_{subgraph} + q^l_{set} + q^l_{trans}). \tag{5.9}$$

VC Dimension Upper Bound. Inspired by the theoretical analysis in [171] that derives an upper bound of the VC dimension for a CNN model, we extend its result to a GCN model, as given by the following lemma.

Lemma 1 Let C^d be the set of GCN models with d convolutional layers. Let $\mathcal{H}^d \triangleq \{h_c : I \to \{0,1\} | c \in C^d\}$ be the set of boolean functions implementable by all GCNs in C^d. The VC dimension of GCNs, as well as CARE, satisfies $\text{VC}_{dim}(\mathcal{H}^d) \leqslant \alpha(d \cdot q(d))^2$ for some constant α. Here, $q(d)$ is the computational complexity of the model under consideration.

VC Dimension Comparison. We now compare the upper bounds of VC dimension on the GCN backbone and CARE, which is formalized by the following theorem.

Theorem 1. Assume that the number of parameters in a GCN backbone and CARE is identical. Let *upperVC(GCN)* and *upperVC(CARE)* be the upper bounds of VC dimension on the two models, respectively, which are given by Lemma 1. We have $upperVC(GCN) > upperVC(CARE)$.

Based on Theorem 1 and VC theory, we conclude that our CARE model exhibits a lower upper bound for the VC dimension compared to its GCN backbone. Consequently, under the condition of an identical number of parameters, GCN augmented with CARE possesses better generalization potential than the original GCN backbone. This theoretical insight suggests that CARE effectively mitigates the generalization issue of GNNs by increasing their upper bounds of the VC dimension.

5.3 Experiments on Brain Networks

We conduct experiments on 5 brain network datasets about 3 different diseases and report the accuracy in Table 5.1. By plugging our proposed CARE into GIN, the performance is improved on all 5 datasets. The improvement is up to 7.8% on ABIDE dataset.

Table 5.1: Brain Classification Results (Average Accuracy ± Standard Deviation). Winner in each backbone/dataset pair is highlighted in **bold**.

Disease	Dataset	Model	
		GIN	CARE-GIN
Parkinson	Taowu	65.00 ± 20.00	**67.50** ± 16.01
	PPMI	57.90 ± 8.12	**59.02** ± 8.55
	Neurocon	**68.50** ± 15.17	**68.50** ± 16.37
Alzheimer	ADNI	61.87 ± 0.38	**62.35** ± 0.61
Autism	ABIDE	57.02 ± 3.88	**61.48** ± 4.09

5.4 Experiments on General Benchmarks

Since our proposed CARE is a plugin component, which does not rely on specific model architecture and dataset. In this section, we further extend CARE to datasets of other domains by plugging it into a large scale of GNNs to verify its universality. The detailed model implementation is presented in Appendix B.2.

5.4.1 Datasets

Nine commonly used benchmark datasets were tested in our experiments. Eight of them were selected from TUDataset [172] and include DD, PROTEINS, MUTAG, NCI1, NCI109, FRANKENSTEIN (FRANK), Tox21 and ENZYMES. The last dataset OGBG-MOLHIV was selected from Open Graph Benchmark [173] and consists of 41K+ graphs. The statistics of the datasets are summarized in Table 5.2.

5.4.2 GNN Backbones

We test the effectiveness of CARE on a wide range of GNN backbones, including GCN [152], GraphSAGE [153], GIN [105], GAT [43], GraphSNN [174], UGformer [168], SAGPool [61],

Table 5.2: Statistics of Datasets.

Dataset	Graph#	Class#	Avg Node#	Avg Edge#
D&D	1178	2	284.32	715.66
PROTEINS	1113	2	39.06	72.82
MUTAG	188	2	17.93	19.79
NCI1	4110	2	29.87	32.30
NCI109	4127	2	29.68	32.13
FRANKENSTEIN	4337	2	16.90	17.88
Tox21	8169	2	18.09	18.50
ENZYMES	600	6	32.63	62.14
OGBG-MOLHIV	41127	2	25.50	27.50

DiffPool [10], HGPSLPool [62], GXN [65]and MEWISPool [175]. We apply CARE to each of them and compare their performance with the original backbone models. Among the 11 models selected, GIN and UGformer are hierarchical ones. We thus apply the hierarchical architecture CARE to them. The global architecture CARE is applied to the rest of the models.

5.4.3 Performance Comparison with GNN Backbones

Effectiveness Analysis. We assess the graph classification performance on the first 8 datasets and the last dataset using two different metrics. The former is assessed by the classification accuracy and the latter by the ROC-AUC. This is because the OGBG-MOLHIV dataset has a severe class imbalance issue. The results on the first 8 datasets are reported in Table 5.3. Each row in the table shows the performance of an original GNN backbone and the performance after applying CARE. Each column reports the results on a dataset. In total there are 88 backbone/dataset pairs and the best result in each pair is highlighted in bold. As shown in Table 5.3, CARE is a clear winner: it outperforms the GNN backbone in 84 out of 88 cases. CARE gains over 1% improvement in the absolute accuracy in 57 out of 88 winning cases, while the drops in the accuracy of all losing cases are all less than 1%. In particular, the improvement of CARE is up to 11.48%, which is achieved on the FRANKENSTEIN dataset with GraphSAGE as backbone. The same observation is made when testing on the OGBG-MOLHIV dataset. As shown in Table 5.4, CARE outperforms the GNN backbones in most cases with improvements up to 5.63%. To sum up, the results demonstrate that CARE is able to serve as a general framework to boost up the graph classification performance over state-of-the-art GNN models

on various datasets. To match with the setting of Theorem 1, we also conduct experiments under the same parameter numbers in Appendix B.3.1. The results demonstrate that CARE can boost up the graph classification performance without introducing additional parameters. A hyperparameter analysis is provided in Appendix B.3.2.

Table 5.3: Graph Classification Results (Average Accuracy ± Standard Deviation). Winner in each backbone/dataset pair is highlighted in **bold**.

Model		D&D	PROTEINS	MUTAG	NCI1	NCI109	FRANK	Tox21	ENZYMES
GCN	Original	71.02 ± 3.17	73.89 ± 2.85	77.52 ± 10.81	78.80 ± 2.01	75.06 ± 2.50	55.58 ± 0.11	88.14 ± 0.29	62.17 ± 6.33
	CARE	**72.15** ± 3.88	**75.01** ± 2.91	**79.30** ± 11.81	**79.66** ± 1.71	**77.39** ± 2.34	**59.67** ± 3.00	**90.59** ± 0.55	**65.00** ± 5.63
GraphSAGE	Original	72.18 ± 2.93	74.87 ± 3.38	75.48 ± 6.11	63.94 ± 2.53	65.46 ± 1.12	52.95 ± 4.01	88.36 ± 0.15	52.50 ± 5.69
	CARE	**73.26** ± 3.25	**76.81** ± 3.30	**81.97** ± 6.42	**76.13** ± 2.03	**75.87** ± 2.51	**64.43** ± 3.15	**90.14** ± 0.74	**55.83** ± 6.88
GIN	Original	73.10 ± 2.44	72.41 ± 4.45	89.36 ± 4.71	81.96 ± 2.03	81.01 ± 1.84	67.23 ± 1.93	92.10 ± 0.59	62.79 ± 7.64
	CARE	**76.32** ± 3.33	**73.14** ± 3.45	**90.47** ± 5.11	**82.34** ± 2.11	**82.15** ± 1.79	**67.33** ± 2.74	**92.43** ± 0.78	**68.17** ± 7.05
GAT	Original	74.25 ± 3.76	74.34 ± 2.09	77.56 ± 10.49	78.07 ± 1.94	74.34 ± 2.18	**62.85** ± 1.90	90.35 ± 0.71	67.67 ± 3.74
	CARE	**75.55** ± 2.43	**76.72** ± 1.74	**79.33** ± 5.82	**78.93** ± 1.69	**76.71** ± 1.45	62.57 ± 2.37	**90.76** ± 0.73	**70.83** ± 5.54
GraphSNN	Original	76.03 ± 2.59	71.78 ± 4.11	84.04 ± 4.09	70.87 ± 2.78	70.11 ± 1.86	**67.17** ± 2.25	92.24 ± 0.59	67.67 ± 3.74
	CARE	**76.67** ± 1.52	**74.02** ± 4.81	**86.71** ± 7.31	**72.25** ± 2.59	**70.46** ± 2.90	66.87 ± 2.33	**92.36** ± 0.58	**68.17** ± 2.94
UGformer	Original	75.51 ± 3.92	70.17 ± 5.42	75.66 ± 8.67	68.84 ± 1.54	66.37 ± 2.74	56.13 ± 2.51	88.06 ± 0.50	64.57 ± 4.53
	CARE	**76.23** ± 4.45	**71.84** ± 3.87	**77.66** ± 5.93	**71.48** ± 2.25	**66.92** ± 1.58	**57.10** ± 2.27	**88.21** ± 0.24	**65.24** ± 5.91
SAGPool	Original	71.46 ± 3.60	74.12 ± 3.46	78.12 ± 8.35	78.34 ± 1.96	76.15 ± 2.25	59.07 ± 2.23	**90.78** ± 0.63	62.00 ± 4.76
	CARE	**73.28** ± 2.25	**74.75** ± 3.14	**79.81** ± 7.52	**79.78** ± 1.67	**76.44** ± 1.74	**59.67** ± 2.04	90.64 ± 0.38	**63.17** ± 4.37
DiffPool	Original	70.45 ± 2.54	72.18 ± 2.80	85.26 ± 4.79	79.78 ± 2.11	76.98 ± 1.88	65.01 ± 3.17	91.02 ± 0.37	48.33 ± 6.67
	CARE	**72.90** ± 4.58	**73.10** ± 3.94	**89.00** ± 7.00	**81.20** ± 2.27	**80.43** ± 1.51	**66.26** ± 2.11	**91.61** ± 0.59	**51.17** ± 6.75
HGPSLPool	Original	71.25 ± 3.25	73.06 ± 3.20	80.82 ± 6.63	79.26 ± 1.44	75.83 ± 1.98	60.82 ± 2.85	90.12 ± 0.47	63.33 ± 5.06
	CARE	**71.61** ± 3.36	**75.47** ± 3.98	**82.31** ± 6.91	**79.77** ± 1.97	**76.87** ± 1.94	**63.36** ± 1.73	**90.44** ± 0.69	**66.00** ± 4.48
GXN	Original	67.62 ± 5.85	70.32 ± 3.03	83.22 ± 7.97	73.34 ± 2.54	72.18 ± 2.24	60.86 ± 2.17	89.93 ± 0.73	63.13 ± 4.68
	CARE	**71.82** ± 4.30	**72.70** ± 2.73	**87.19** ± 6.61	**74.75** ± 2.90	**73.78** ± 1.66	**62.64** ± 2.27	**90.43** ± 0.76	**64.44** ± 6.96
MEWISPool	Original	**76.03** ± 2.59	68.10 ± 3.97	84.73 ± 4.73	74.21 ± 3.26	75.30 ± 1.45	64.63 ± 2.83	88.13 ± 0.05	53.66 ± 6.07
	CARE	75.72 ± 2.54	**69.64** ± 3.69	**86.70** ± 4.27	**76.48** ± 2.74	**75.34** ± 2.86	**67.79** ± 2.34	**88.65** ± 0.07	**55.67** ± 6.33

Table 5.4: Graph Classification Results (Average ROC-AUC ± Standard Deviation) on OGBG-MOLHIV dataset. Winner in each backbone/dataset pair is highlighted in **bold**.

	GraphSAGE	GCN	GIN	GAT	GXN
Original	70.37 ± 0.42	73.49 ± 1.99	65.11 ± 2.56	75.83 ± 1.78	69.15 ± 0.01
CARE	**74.33** ± 2.12	**74.29** ± 1.07	**65.42** ± 3.70	**76.89** ± 2.18	**69.17** ± 0.02
	UGformer	SAGPool	DiffPool	HGPSLPool	MEWISPool
Original	77.23 ± 3.54	73.80 ± 1.86	**71.63** ± 2.25	76.08 ± 2.86	**79.66** ± 1.71
CARE	**78.04** ± 3.19	**74.42** ± 1.53	70.21 ± 2.79	**77.23** ± 2.16	77.37 ± 1.05

Generalization Performance. We also observe that CARE is able to alleviate the overfitting in GNN backbones. An example is shown in Fig. 5.3, where we plot the accuracy curves on the PROTEINS dataset with GCN as the backbone. It shows that the test accuracy of GCN (in blue) exhibits a steep and continuous downward trend starting from epoch 120, while its corresponding training accuracy continues to climb up. This indicates an obvious overfit of

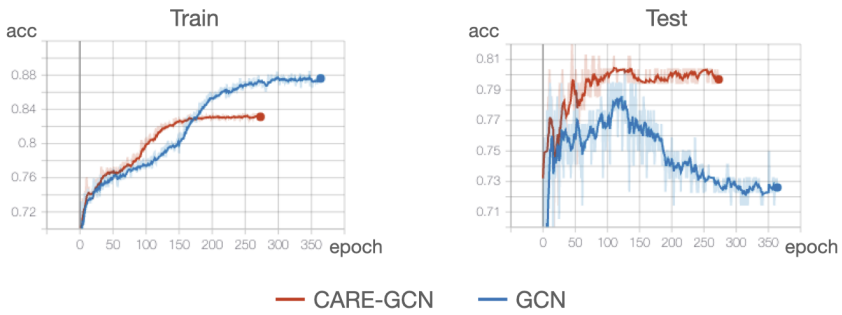


Fig. 5.3: Accuracy curves of CARE-GCN and GCN on PROTEINS dataset.

GCN to the training data. After applying CARE on GCN (in red), the steep drop in the test accuracy vanishes, which demonstrates the ability of CARE in remitting overfitting.

5.4.4 Ablation Studies

We first conduct experiments to verify the effectiveness of our proposed Class-Aware Refiner. Then we perform two ablation studies to show how different designs of the Subgraph Selector and the loss function $\mathcal{L}$ influence the performance of CARE. We also evaluate two different choices for the similarity metric used in the class loss.

Class-Aware Refiner. To evaluate the effectiveness of the Class-Aware Refiner, we remove it from the framework and maintain the subgraph selector. To replace Eq. (3), the subgraph representation is used to refine the original graph representation:

$$\boldsymbol{hg}'_G = \text{Trans}([\boldsymbol{hg}_G \mid \boldsymbol{hg}_G^{sub}]), \tag{5.10}$$

The results reported in Table 5.5 demonstrate that the classification accuracy decreases in most cases when only using the subgraph selector without Class-Aware Refiner. This observation indicates that the class representations obtained from the refiner can better enhance the graph representations.

Subgraph Selector. We compare the performance of three CARE variants with different subgraph selectors on 5 datasets with 4 GNN backbones. The results are presented in Table 5.6. The first CARE variant, denoted as "None", uses the whole graph for class representation learning without selecting any subgraph. The other two models apply SAGPool [61] and HGPSLPool [62] respectively as the subgraph selector. The pooling ratio for both SAGPool and

Table 5.5: Ablation Study on Class-Aware Refiner. Winner in each backbone/dataset pair is highlighted in bold.

	Refiner	D&D	PROTEINS	MUTAG	NCI1	NCI109
GraphSAGE	w/o	72.37 ± 3.56	75.47 ± 3.72	76.58 ± 7.28	75.08 ± 2.19	73.08 ± 2.53
	w/	**73.26** ± 3.25	**75.92** ± 2.84	**81.97** ± 6.42	**75.23** ± 1.76	**73.58** ± 1.68
GCN	w/o	71.53 ± 3.66	74.09 ± 3.91	**80.88** ± 7.45	79.29 ± 2.35	75.25 ± 2.51
	w/	**72.15** ± 3.88	**75.01** ± 2.91	79.30 ± 11.81	**79.66** ± 1.71	**75.75** ± 1.63

HGSPLPool is set to 0.5. We can see that using the subgraph selector achieves the best result in 15 out of 20 cases. When comparing the performance of using SAGPool and HGSPLPool, the former beats the latter in 14 out of 20 cases. Therefore, we choose SAGPool as the default subgraph selector.

Table 5.6: Ablation Study on Different Subgraph Selectors. Winner is highlighted in **bold**.

Backbone	Subgraph Selector	D&D	PROTEINS	MUTAG	NCI1	NCI109
GraphSAGE	None	67.24 ± 4.64	75.01 ± 4.15	**82.95** ± 5.86	**77.66** ± 1.98	73.67 ± 1.28
	SAGPool	**73.26** ± 3.25	**75.92** ± 2.84	81.97 ± 6.42	75.23 ± 1.76	73.58 ± 1.68
	HGPSLPool	71.82 ± 3.99	75.28 ± 3.76	77.57 ± 7.33	75.84 ± 1.50	**74.36** ± 2.17
GCN	None	71.05 ± 3.89	73.39 ± 3.45	78.74 ± 9.67	79.15 ± 1.66	**76.30** ± 2.31
	SAGPool	**72.15** ± 3.88	**75.01** ± 2.91	**79.30** ± 11.81	**79.66** ± 1.71	75.75 ± 1.63
	HGPSLPool	68.75 ± 3.45	73.39 ± 3.45	77.66 ± 5.13	79.25 ± 1.90	75.35 ± 2.57
GIN	None	**75.64** ± 3.38	71.96 ± 5.49	88.80 ± 5.05	**82.85** ± 1.27	82.11 ± 1.60
	SAGPool	74.70 ± 3.37	72.32 ± 4.25	90.44 ± 4.58	82.34 ± 2.11	**82.15** ± 1.79
	HGPSLPool	75.06 ± 3.49	**72.86** ± 4.66	**90.47** ± 6.10	81.63 ± 1.80	81.05 ± 1.10
GAT	None	74.28 ± 2.43	75.74 ± 2.88	78.22 ± 6.77	75.30 ± 3.01	74.90 ± 2.24
	SAGPool	**75.38** ± 2.93	**76.72** ± 1.74	77.69 ± 8.99	**78.52** ± 2.12	**76.39** ± 2.76
	HGPSLPool	74.79 ± 2.73	75.46 ± 3.56	**79.33** ± 9.73	75.74 ± 2.32	74.15 ± 3.53

Class Loss. We investigate different designs of the loss function to study the impact of the class loss we proposed. We consider four designs of the overall loss function $\mathcal{L}$: a) *cls*: uses the classification loss $\mathcal{L}_{cls}$ only; b) *intra*: uses a combination of the classification loss and the intra-class loss as given by $\mathcal{L} = \mathcal{L}_{cls} - \lambda_2 * \exp(\mathcal{L}_{intra})$; c) *inter*: uses a combination of the classification loss and the inter-class loss as given by $\mathcal{L} = \mathcal{L}_{cls} + \lambda_2 * \exp(\mathcal{L}_{inter})$; and d) *combine*, the overall loss function in Eq. (5.7). The results in Table 5.7 show that the proposed loss function performs the best among all designs. This demonstrates the effectiveness of the proposed class loss that takes into account the intra-class similarity and inter-class discrepancy.

Similarity Metric for Class Loss. We study the effect of the similarity metric in computing the class loss. Besides the cosine similarity, L2 distance is also commonly used to measure

Table 5.7: Ablation Study on Design of Loss Function in terms of Classification Accuracy. Winner in each backbone/dataset pair is highlighted in **bold**.

Backbone	Loss	D&D	PROTEINS	MUTAG
Graph SAGE	*cls*	72.24 ± 2.72	75.82 ± 3.34	75.50 ± 6.05
	intra	70.12 ± 3.27	75.83 ± 2.83	77.02 ± 10.45
	inter	70.30 ± 3.61	75.02 ± 3.55	81.87 ± 7.67
	combine	**73.26** ± 3.25	**75.92** ± 2.84	**81.97** ± 6.42
GCN	*cls*	71.39 ± 2.81	74.21 ±2.74	77.11 ± 9.48
	intra	71.89 ± 5.35	74.65 ± 4.10	78.22 ± 7.17
	inter	71.31 ± 2.06	74.20 ± 3.88	78.18 ± 10.41
	combine	**72.15** ± 3.88	**75.01** ± 2.91	**79.30** ± 11.81

the dissimilarity between two vectors. We use L2 distance in place of the cosine similarity in Eqs. (5.5) and (5.6) to define the intra-class and inter-class losses. As L2 distance measures the dissimilarity, we take an inverse of the class loss when L2 distance is used. We then compare the performance of CARE under these two different metrics. Table 5.8 shows the results under GCN as the GNN backbone. It shows that CARE with the cosine similarity is more powerful than that with the L2 distance.

$$\mathcal{L}_{intra} = \text{norm}(\sum_{i=1}^{|\mathcal{Y}|} \text{norm}(\sum_{y_G=i} \text{dis}(\boldsymbol{hc}_i, \boldsymbol{hg}_G^{sub}))), \tag{5.11}$$

$$\mathcal{L}_{inter} = \text{norm}(\sum_{i=1}^{|\mathcal{Y}|} \text{norm}(\sum_{j=i}^{|\mathcal{Y}|} \text{dis}(\boldsymbol{hc}_i, \boldsymbol{hc}_j))), \tag{5.12}$$

$$\mathcal{L}_{class} = \exp(\mathcal{L}_{intra} - \mathcal{L}_{inter}). \tag{5.13}$$

Table 5.8: Ablation Study on Similarity Metric for Class Loss.

	D&D	PROTEINS	MUTAG
-	71.39 ± 2.81	74.21 ± 2.74	77.11 ± 9.48
L2	71.91 ± 4.97	74.65 ± 3.54	77.69 ± 8.35
KL	71.85 ± 4.49	74.38 ± 3.16	78.74 ± 5.19
cos_sim	**72.15** ± 3.88	**75.01** ± 2.91	**79.30** ± 11.81

5.4.5 Case Study for Class Separability

We design a case study to further investigate the effect of CARE, in particular its class-aware components, in refining graph representations for the classification task. The idea is to study how CARE affects the separability of graph classes in the training process. Is it able to direct the graph representation learning to move towards better class separability? In order to answer this question, we use four class separability metrics as follows. For all metrics, the larger their values, the better the class separability is. We refer the reader to Appendix B.4 for their formal definitions.

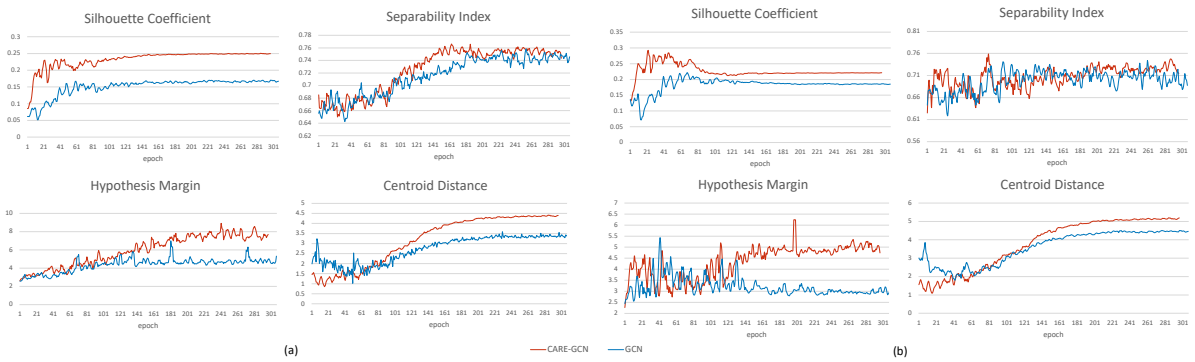


Fig. 5.4: (a) Class Separability on PROTEINS with GCN Backbone (Training Set). (b) Class Separability on PROTEINS with GCN Backbone (Test Set). The results were obtained by passing the test data once at the end of each training epoch. Note that this process doesn't affect the training in any way as the model parameters/loss are not updated when passing the test data.

Silhouette Coefficient [176]. It measures how similar a sample is to its own class (cohesion) compared to those from other classes (separation). Its value ranges from -1 to 1.

Separability Index [177]. It computes the fraction of samples that have a nearest neighbour with the same class label. Its value ranges from 0 to 1.

Hypothesis Margin [178]. It measures the distance between a sample's nearest neighbor from the same class (near-hit) and the nearest neighbor from the opposing class (near-miss) and averages over all samples.

Centroid Distance. It sums up the distances between the centroids for all pairs of classes, where the centroid of a class is the mean of all samples in the class.

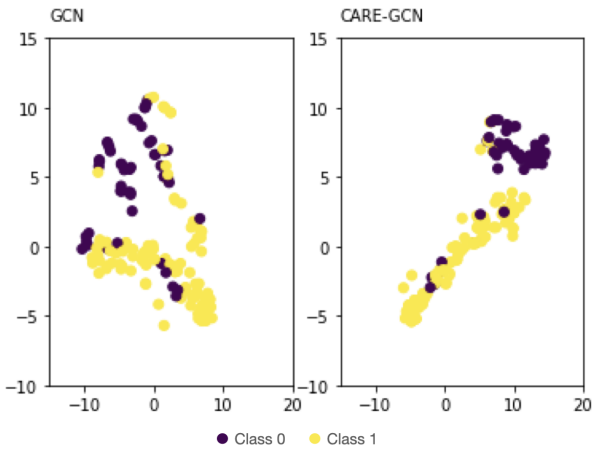


Fig. 5.5: Visualization of Graph Representations Produced by GCN and CARE-GCN on PROTEINS dataset.

Fig. 5.4 reports the results on the PROTEINS dataset with GCN as the backbone. We compute the four metrics on the graph representations produced by CARE and GCN on the training data. CARE uses the refined graph representations, while GCN uses the original ones. It can be seen that the training curves of CARE exhibit an upward trend under all the four separability metrics. At the time when models converge, CARE outperforms GCN in all metrics. In particular, CARE achieves 49.26% improvement on Silhouette Coefficient, 1.96% on Separability Index, 45.21% on Hypothesis Margin, and 30.51% on centroid distance. A visualization of the graph representations in each model is shown in Fig. 5.5. Graph representations are passed into T-SNE [1] for dimensionality reduction and colored by their class labels. This demonstrates that CARE is indeed able to steer the graph representation learning towards better class separability, which is also reflected by its superior classification performance over GNN backbones. Similar conclusions can be drawn from the results on the test data, which indicates that the class separability of CARE generalizes well to the test data.

5.4.6 Time Efficiency

CARE, when applied to a GNN backbone, introduces an additional refiner and the class loss. A natural question arises: will CARE significantly sacrifice the efficiency of its GNN backbone for better classification performance? This subsection aims to answer this question. Table 5.9 reports the number of epochs and the total time needed (including training, validation and test)

for CARE and the backbones GraphSAGE and GCN. It can be seen that CARE takes less number of epochs to converge than its GNN counterpart in all cases. Consequently, the running time of CARE is shorter than (4 out of 6 cases) or comparable to its backbones. The results demonstrate that CARE is able to work on top of existing GNN models with superior effectiveness and improved/comparable efficiency, making it a practical choice in real applications.

Table 5.9: Time Efficiency of CARE and Backbones. Total time (h) was recorded for a single run (including training, validation, and test) with batch size 20 and 10-fold CV. Best time in each backbone/dataset pair is highlighted in bold.

Model		D&D		PROTEINS		MUTAG	
		Epoch # ± s.d.	Time	Epoch # ± s.d.	Time	Epoch # ± s.d.	Time
GraphSAGE	Original	500.6 ± 123.2	1.205	320.5 ± 53.2	1.209	384.1 ± 101.3	0.180
	CARE	293.5 ± 12.0	**1.142**	282.0 ± 51.5	**0.911**	302.2 ± 34.1	**0.159**
GCN	Original	267.4 ± 3.4	**0.692**	365.0 ± 27.9	0.670	352.4 ± 69.9	**0.132**
	CARE	264.1 ± 5.2	0.848	306.5 ± 17.2	**0.665**	332.4 ± 57.3	0.143

5.5 Summary

In this section, we proposed CARE, a novel graph representation refinement framework for GNN-based graph classification. It fills the gap that existing GNNs fail to consider graph-level relationships in model design, and meanwhile improves the model generalization as proven theoretically and evidenced empirically. Its plug-in-play nature makes it a powerful framework to build upon arbitrary GNN models and boost up their classification performance without sacrificing efficiency.

Chapter 6

Contrastive Graph Pooling

In this chapter[1], we propose a contrastive dual-attention block and a differentiable graph pooling method called *ContrastPool* to better utilize GNN for brain networks, meeting fMRI-specific requirements. We apply our method to 5 resting-state fMRI brain network datasets of 3 diseases and demonstrate its superiority over state-of-the-art baselines. Our case study confirms that the patterns extracted by our method match the domain knowledge in neuroscience literature, and disclose direct and interesting insights. Our contributions underscore the potential of ContrastPool for advancing the understanding of brain networks and neurodegenerative conditions.

6.1 Data Characteristics of fMRI

Compared with conventional machine learning models that handle vector-based data, GNNs are able to engage graph topological information through message passing. In light of the promising performance of GNNs on other applications, several studies [23–25] have applied GNNs to brain network analysis. However, directly applying general-purpose GNNs to brain networks could be ineffective on fMRI data due to its unique data characteristics [26].

1 **Low signal-to-noise ratio.** fMRI is a type of high dimensional data, while non-neural noise derived from cardiac/respiratory processes or scanner instability could cause large variations within a single subject and across different subjects.

[1]The work in this chapter has been published as "Contrastive Graph Pooling for Explainable Classification of Brain Networks", IEEE Transactions on Medical Imaging, 2024. [179]

2 **Node alignment.** Different from other graph-based datasets, each node of a brain network represents an ROI under a brain parcellation scheme. If the same scheme is applied to all subjects, each resulting network will have the same number of nodes and the nodes are aligned across different subjects.

3 **Limited data scale.** Due to the limited availability of the fMRI data, brain network datasets contain a relatively small number of subjects, which can cause overfitting for GNNs.

To better utilize GNNs for brain networks and meet the need of fMRI characteristics, we propose a contrastive graph pooling method (*ContrastPool*) with a contrastive dual-attention to extract group-specific information. The ROI-wise attention is introduced to identify the most discriminative regions and filter out noise. The subject-wise attention utilizes the characteristic of node alignment for brain networks. With group aggregation, we obtain a contrast graph and use it to guide message passing. The group knowledge can weaken the sensitivity to specific subjects to remit overfitting.

6.2 Methodology

This chapter propose ContrastPool, which addresses the above challenges by aggregating subjects over different classes using dual-attention. The idea of ContrastPool is based on an important observation: the relatedness of different ROIs for different diseases varies, and the extent to which a subject exhibits typical characteristics of a disease also varies. Therefore, we design ContrastPool in a way that the graphs in the same group (e.g., ASD group) are summarized by assigning different weights for ROIs and subjects. These weights are not assigned manually but learnt automatically from training data to optimize the classification performance. The architecture of ContrastPool is shown in Fig. 6.1, with Autism as an example. In the training stage, all training graphs are categorized by their groups and pass through the Contrastive Dual-Attention (CDA) block to generate the contrast graph (the upper block of Fig. 6.1). This contrast graph is subsequently encoded into an assignment matrix, which is combined with each input graph for hierarchical graph pooling (the lower block of Fig. 6.1). In the validation and test stages, each validation/test graph passes through the lower block and is fused with

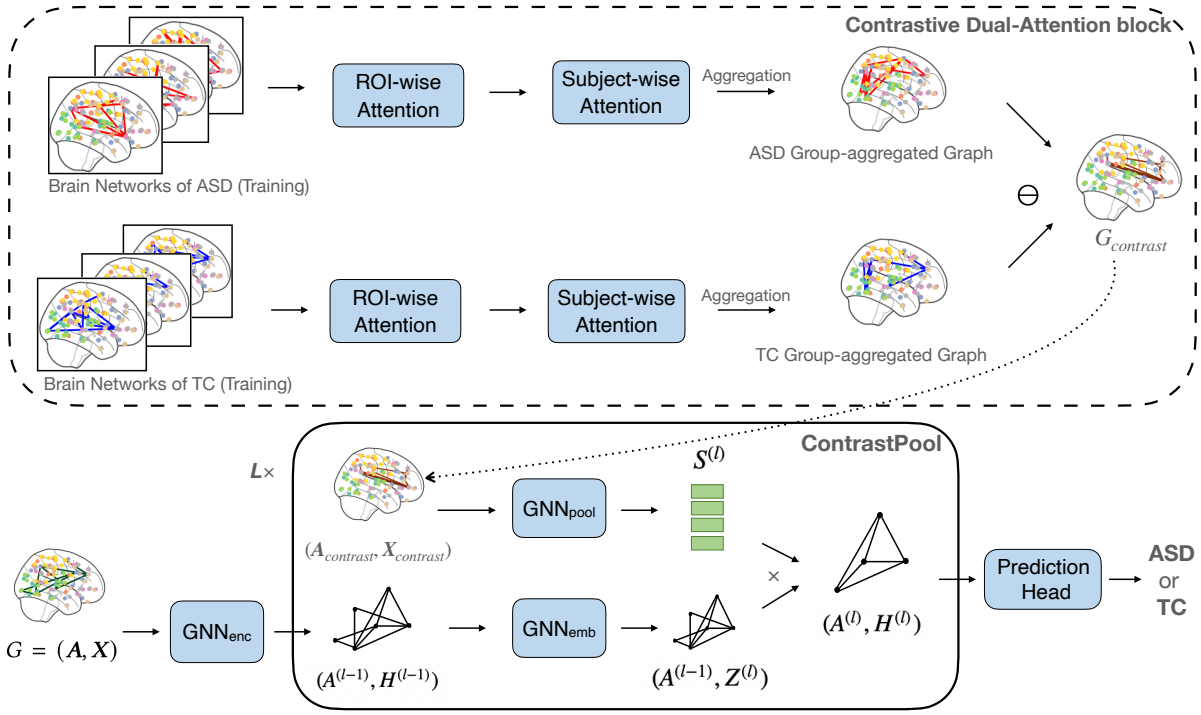


Fig. 6.1: The architecture of ContrastPool, using Autism as an example.

the contrast graph (generated by the training phase) to obtain its graph representation. This representation is then fed into the Prediction Head to produce the final prediction of the graph.

In the following, we first introduce the Contrastive Dual-Attention block, which is used to generate a contrast graph. We then describe how the ContrastPool module produces node and graph representations based on the contrast graph. Finally, we discuss our design of the loss function.

6.2.1 Contrastive Dual-Attention (CDA) Block

This block aims to generate a contrast graph that best characterizes the differences in brain networks between two groups of subjects. In order to achieve this, it first summarizes all the graphs within the same group into a summary graph. It then computes the difference of the summary graphs from two different groups. This chapter designs the computation between the summary graph as a learnable dual-attention process such that the most discriminative ROIs and the most representative subjects can be automatically highlighted in the summary graph formation.

Consider two groups of graphs in the training set, $\mathcal{G}^{TC} = (\mathcal{A}^{TC}, \mathcal{X}^{TC})$ and $\mathcal{G}^{ASD} = (\mathcal{A}^{ASD}, \mathcal{X}^{ASD})$, bearing Typical Control (TC) and Autism Spectrum Disorder (ASD) groups, respectively. CDA first computes the summary graphs G^{TC}_{sum} and G^{ASD}_{sum} of the two groups via two subject aggregation functions $\mathrm{SA}_A(\cdot)$ and $\mathrm{SA}_X(\cdot)$ performed respectively on the adjacency matrix set $\mathcal{A}^{TC}$, $\mathcal{A}^{ASD}$ and the feature matrix set $\mathcal{X}^{TC}$, $\mathcal{X}^{ASD}$:

$$G^{TC}_{sum} = (\boldsymbol{A}^{TC}_{sum}, \boldsymbol{X}^{TC}_{sum}) = (\mathrm{SA}_A(\mathcal{A}^{TC}), \mathrm{SA}_X(\mathcal{X}^{TC})), \tag{6.1}$$

$$G^{ASD}_{sum} = (\boldsymbol{A}^{ASD}_{sum}, \boldsymbol{X}^{ASD}_{sum}) = (\mathrm{SA}_A(\mathcal{A}^{ASD}), \mathrm{SA}_X(\mathcal{X}^{ASD})). \tag{6.2}$$

The contrast graph $G_{contrast} = (\boldsymbol{A}_{contrast}, \boldsymbol{H}_{contrast})$ is then obtained by:

$$\boldsymbol{A}_{contrast} = \boldsymbol{A}^{TC}_{sum} \ominus \boldsymbol{A}^{ASD}_{sum}, \tag{6.3}$$

$$\boldsymbol{H}_{contrast} = \boldsymbol{X}^{TC}_{sum} \ominus \boldsymbol{X}^{ASD}_{sum}, \tag{6.4}$$

where $\ominus$ is a binary function that computes the element-wise absolute differences of two matrices.

To enable the automatic learning of the weights on the ROIs and subjects, respectively, we design a dual-attention function as $\mathrm{SA}(\cdot)$. In general, a self-attention function [72] for a given matrix $\boldsymbol{X} \in \mathbb{R}^{k \times m}$ can be written as:

$$\mathrm{Attn}(\boldsymbol{X}) = \mathrm{norm}\left(\boldsymbol{X} + \phi\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathsf{T}}}{\sqrt{k}}\right)\boldsymbol{V}\right), \tag{6.5}$$

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K, \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V, \tag{6.6}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{m \times m}$ are parameter matrices, $\phi(\boldsymbol{z})_i = \mathrm{softmax}(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{m} e^{z_j}}$, for $i = 1 \ldots k, \boldsymbol{z} \in \mathbb{R}^k$, and $\mathrm{norm}(\cdot)$ is layer normalization function. Essentially, the dual-attention mechanism is performed by feeding different input matrices to Eq. (6.5) for different levels of attention. We illustrate the dual-attention process with $\mathcal{X}^{TC}$ as an example below. We first apply an ROI-wise attention to compute the attention between the ROIs in each subject. That is, the input matrix for the ROI-attention function $\mathrm{Attn}_{ROI}(\cdot)$ is the adjacency matrix of each subject

with $k = m$. On top of this ROI-wise attention, we apply subject-wise attention to compute the attention between subjects in each ROI. The input matrix for the subject-wise attention function $\text{Attn}_{subject}(\cdot)$ is the subject-stacked resultant matrix of each ROI, with a dimension of $n^{TC} \times d$, where n^{TC} is the number of subjects in $\mathcal{X}^{TC}$. Finally, the output of the subject-wise attention is averaged on the subject dimension to obtain the summary feature matrix. The aggregation function $\text{SA}_X(\cdot)$ performed on $\mathcal{X}^{TC}$ can be formally written as:

$$\boldsymbol{T}^{TC}_{ROI} = \left[\text{Attn}_{ROI}(\boldsymbol{X}) : \boldsymbol{X} \in \mathcal{X}^{TC}\right], \tag{6.7}$$

$$\boldsymbol{T}^{TC}_{subject} = \left[\text{Attn}_{subject}\left(\boldsymbol{T}^{TC}_{ROI}(:, i, :)\right) : i = 1, \ldots, m\right], \tag{6.8}$$

$$\text{SA}_X(\mathcal{X}^{TC}) = \frac{\sum_{j=1}^{n^{TC}} \boldsymbol{T}^{TC}_{subject}(j, :, :)}{n^{TC}}, \tag{6.9}$$

where $\boldsymbol{T}^{TC}_{ROI}, \boldsymbol{T}^{TC}_{subject} \in \mathbb{R}^{n^{TC} \times m \times m}$ denote two 3-dimensional matrices to store the outputs of the ROI-wise and subject-wise attention functions, and $[\cdot]$ denotes the stack operation to combine a set of 2D matrices to a single 3D matrix. The subject aggregation function $\text{SA}_A(\cdot)$ performed on the adjacency matrix sets can be defined in a similar way.

The ROI-wise attention in Eq. (6.7) aims to extract discriminative-related ROIs and to mitigate the influence of noise (caused by factors such as cardiac/respiratory noise or scanner instability). We do an empirical study of how ROI-wise attention reduces site-specific noise in Section 6.3.3. The goal of subject-wise attention in Eq. (6.8) is to leverage the node alignment and focus on the most representative subjects in each class. Utilizing ROI- and subject-wise attention to filter crucial information from two different aspects enables us to extract features more effectively from brain networks. This design of dual-attention can also guarantee the permutation invariance of brain networks, which means no matter how we permute the indices of subjects or ROIs, the output will be the same. By averaging over all subjects from the same group to compute the summary graph via Eq. (6.9), the impact of the number of subjects in different groups is alleviated. This helps mitigate the class imbalance issue in some datasets (e.g., PPMI and ADNI). The contrast between summary graphs can also weaken the sensitivity to the non-representative subjects and thus mitigate overfitting.

6.2.2 Pooling with a Contrast Graph

This module aims to produce a high-quality representation for each input graph by leveraging the group-discriminative information captured in the contrast graph. To achieve a natural and effective utilization of the contrast graph in graph pooling and meanwhile extract high-level node features, we adopt DiffPool [10] as the pooling method. The idea is to coarsen the input graph in a hierarchical manner (via layers) such that similar nodes are grouped together into clusters at each layer to extract high-level node representations. Specifically, our ContrastPool takes the input graph with m number of nodes and coarsens it by grouping the input nodes into e.g., $\frac{m}{2}$ number of clusters in a soft manner. The node grouping is performed on an embedded node feature matrix $\boldsymbol{Z}$ and guided by a cluster assignment matrix $\boldsymbol{S}$ that characterizes the node similarity. The output coarsened graph after this pooling layer contains $\frac{m}{2}$ number of nodes, with each node representing a soft cluster of the input nodes. For hierarchical pooling, L pooling layers can be deployed.

In ContrastPool, the cluster assignment matrix can be naturally implemented by the contrast graph as it captures the relatedness of ROIs to the prediction task. Incorporating the contrast graph into graph pooling as group-based prior knowledge can also alleviate overfitting for datasets with limited scales. As shown in Eq. (6.10), the contrast graph $G_{contrast} = (\boldsymbol{A}_{contrast}, \boldsymbol{X}_{contrast})$ passes through a GNN pooling, $\text{GNN}^{(l)}_{pool}$, to learn a cluster assignment matrix $\boldsymbol{S}^{(l)} \in \mathbb{R}^{m^{(l-1)} \times m^{(l)}}$. Herein, $m^{(l)}$ is a pre-defined number of clusters at layer l controlled by a hyperparameter named the pooling ratio.

$$\boldsymbol{S}^{(l)} = \text{softmax}\left(\text{GNN}^{(l)}_{pool}(\boldsymbol{A}_{contrast}, \boldsymbol{X}_{contrast})\right). \tag{6.10}$$

To obtain the embedded node feature matrix $\boldsymbol{Z}^{(l)}$ at layer l, we pass the output graph $G^{(l-1)} = (\boldsymbol{A}^{(l-1)}, \boldsymbol{H}^{(l-1)})$ from the previous layer through an embedding GNN:

$$\boldsymbol{Z}^{(l)} = \text{GNN}^{(l)}_{emb}(\boldsymbol{A}^{(l-1)}, \boldsymbol{H}^{(l-1)}), \tag{6.11}$$

where the initial representation $\boldsymbol{H}^{(0)}$ is the input feature matrix encoded by a GNN, i.e., $\boldsymbol{H}^{(0)} = \text{GNN}_{enc}(\boldsymbol{X})$.

Once we obtain the cluster assignment matrix $\boldsymbol{S}^{(l)}$ and the embedded node feature matrix $\boldsymbol{Z}^{(l)}$, we generate a coarsened adjacency matrix $\boldsymbol{A}^{(l)}$ and a new feature matrix $\boldsymbol{H}^{(l)}$. This coarsening process can reduce the number of nodes to get higher-level node representations. In particular, we apply the following two equations:

$$\boldsymbol{H}^{(l)} = \boldsymbol{S}^{(l)^\top} \boldsymbol{Z}^{(l)} \in \mathbb{R}^{m^{(l)} \times d}, \tag{6.12}$$

$$\boldsymbol{A}^{(l)} = \boldsymbol{S}^{(l)^\top} \boldsymbol{A}^{(l-1)} \boldsymbol{S}^{(l)} \in \mathbb{R}^{m^{(l)} \times m^{(l)}}. \tag{6.13}$$

Through Eq. (6.12 and 6.13), ContrastPool is able to produce high-quality representations for each input graph by extracting high-level node representations under the guidance of the contrast graph. Intuitively, if an edge with two end ROIs v_1 and v_2 has a high weight in $\boldsymbol{A}_{contrast}$ (i.e., the dual attention of this connection differs significantly in the two groups), the message passing between v_1 and v_2 will be more pronounced. As a result, the scores of v_1 and v_2 in $\boldsymbol{S}$ will share more similarity and contribute to the set of clusters in a similar manner. With the help of the entropy loss $\mathcal{L}_{E_1}$ (to be introduced in the subsection of "Loss Function"), the learning of the assignment scores of a node will be guided to favor only a few clusters. As a result, two ROIs whose connection is highlighted by the contrast graph will be grouped into the same cluster and high-level node features will be extracted.

6.2.3 Loss Function.

Technically, an assignment matrix with entries close to uniform distribution could guide the GNN to treat each ROI and subsequently each node cluster produced in hierarchical pooling equally. Thus, besides the commonly-used cross-entropy loss $\mathcal{L}_{cls}$ [167] for graph classification, we also adopt the loss in [10] to regularize the entropy of the assignment matrix to avoid such equal treatment. Since a fully-connected adjacency matrix of the contrast graph could cause the over-smoothing of GNN, we apply an entropy loss to the adjacency matrix of the contrast graph $\boldsymbol{A}_{contrast}$ to sparsify the matrix. Note that imposing an entropy loss on $\boldsymbol{A}_{contrast}$ is different from thresholding the edges of $\boldsymbol{A}_{contrast}$. The former has an active impact on the learning of $\boldsymbol{A}_{contrast}$ towards the sparse formation, while the latter is a post-process on an already learnt matrix. The two entropy losses shown in Eq. (6.14 and 6.15) can help class-indicative ROIs to stand out and subsequently boost the model performance.

$$\mathcal{L}_{E_1} = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{m^{(l)}}\sum_{i=1}^{m^{(l)}}\text{entropy}\left(\boldsymbol{S}^{(l)}(i,:)\right), \tag{6.14}$$

$$\mathcal{L}_{E_2} = \frac{1}{m}\sum_{i=1}^{m}\text{entropy}\left(\boldsymbol{A}_{contrast}(i,:)\right). \tag{6.15}$$

The overall loss $\mathcal{L}$ of ContrastPool is defined as Eq. (6.16), where λ_1 and λ_2 are trade-off hyperparameters for balancing different losses.

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 * \mathcal{L}_{E_1} + \lambda_2 * \mathcal{L}_{E_2}. \tag{6.16}$$

6.3 Experimental Study

In this section, we first introduce baselines we used, followed by assessing the performance of ContrastPool in comparison with these models. We then present three case studies to show how ContrastPool meets fMRI-specific requirements, and meanwhile provide the domain interpretation of our dual-attention mechanism. We further conduct ablation studies to analyze the effects of the components in our model.

6.3.1 Baseline Models

We select various models as baselines, including (1) conventional machine learning models: Logistic Regression, Naïve Bayes, Support Vector Machine Classifier (SVM), k-Nearest Neighbours (kNN) and Random Forest (implemented by the scikit-learn library [180]); (2) general-purposed GNNs: GCN [152], GraphSAGE [153], GIN [105], GAT [43] and GatedGCN [154]; (3) typical graph pooling approaches, DiffPool [10], SAGPool [61], HGPSLPool [62] and MEWISPool [175]; (4) dense contrast graph with SVM: cs [2]; and (5) neural networks designed for brain networks: BrainNetCNN [20], LiNet [79], PRGNN [24], MG2G [80], BrainGNN [25] and BNTF [77]. For GNN baseline models, we sparsify all input graphs by keeping 20% edges with top correlations in $\boldsymbol{A}$, to avoid over-smoothing. The implementation detail of our experiments is given in Appendix C.1.

Table 6.1: Graph Classification Results (Average Accuracy ± Standard Deviation) over 10-fold-CV. The best result is highlighted in **bold**. The second best result is underlined.

	Model	Taowu	PPMI	Neurocon	ADNI	ABIDE
Conventional ML methods	Logistic Regression	**77.50**±7.50	56.50±11.02	68.50±15.17	61.99±0.59	65.82±3.51
	Naïve Bayes	65.00±12.25	58.83±5.42	63.50±11.84	48.27±4.44	63.50±2.69
	SVM	65.00±16.58	63.67±5.11	71.50±13.93	61.77±0.25	60.67±3.61
	kNN	62.50±12.50	53.52±10.34	56.00±21.77	62.59±1.73	60.37±5.64
	Random Forest	57.50±22.50	62.23±4.22	58.50±11.19	61.77±0.25	61.18±5.01
General-purposed GNNs	GCN	60.00±29.15	54.02±9.06	59.00±20.71	61.57±0.60	60.97±2.84
	GraphSAGE	60.00±33.91	55.00±12.89	68.50±15.17	61.19±1.72	63.09±3.11
	GIN	65.00±20.00	57.90±8.12	68.50±15.17	61.87±0.38	57.02±3.88
	GAT	67.50±22.50	54.98±8.03	54.00±15.62	61.34±1.27	60.87±5.02
	GatedGCN	65.00±22.91	52.60±11.51	69.00±25.48	62.06±4.53	63.60±4.70
	DiffPool	65.00±27.84	58.00±11.00	62.50±25.62	63.80±4.64	63.75±3.16
	SAGPool	65.00±25.50	52.14±11.69	65.50±16.95	58.37±3.89	63.70±3.76
	HGPSLPool	60.00±30.00	52.57±11.04	68.50±21.91	59.71±5.79	63.60±3.50
	MEWISPool	57.50±22.50	54.07±12.15	66.00±15.94	61.78±3.10	63.99±4.48
Models for brain networks	cs	**77.50**±17.50	58.36±4.40	63.50±11.84	62.25±0.47	62.59±3.14
	BrainNetCNN	65.00±27.84	57.33±10.32	66.00±22.45	61.08±2.87	65.86±2.36
	LiNet	55.00±24.49	60.71±10.61	56.00±26.91	63.64±1.73	58.14±3.72
	PRGNN	67.50±31.72	58.83±6.89	63.00±23.37	60.71±2.21	60.76±4.12
	MG2G	57.50±22.50	55.45±10.24	68.00±19.77	63.64±5.10	64.41±2.16
	BrainGNN	67.50±25.12	61.71±6.05	56.50±23.03	61.05±1.23	62.88±2.46
	BNTF	65.00±21.08	51.60±6.15	66.00±11.97	61.94±3.11	63.70±4.84
Ours	ContrastPool	**77.50**±17.50	**64.00**±6.63	**75.00**±15.81	**67.08**±2.63	**68.63**±2.65

6.3.2 Comparison with Baselines

We report the accuracy on 5 brain network datasets in Table 6.1. Compared with conventional ML methods, general-purposed GNNs and models for brain network do not show a significant advantage and attain similar performance in most cases. This finding is consistent with the results reported in existing papers [25]. Our ContrastPool outperforms all 21 GNN and ML baselines on all datasets. In particular, ContrastPool improves over all GNNs specifically designed for brain networks by up to 13.6%. The average improvement over GNNs is 8.39%. The p-values of one-sided paired t-tests comparing our ContratPool with the best model of brain networks on two large datasets, ADNI and ABIDE, are 0.0483 and 0.00798, respectively. This indicates that our model significantly outperforms existing methods on these two datasets. However, on small datasets with large standard deviations in 10 folds, t-tests are highly sensitive to outliers existing in the 10-fold results, thus decreasing the t-statistic calculated and

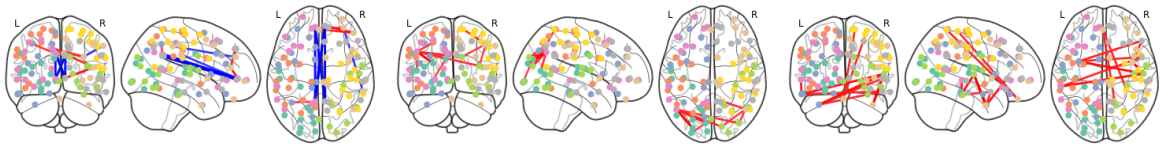
Table 6.2: Results of more evaluation metrics on Taowu, Neurocon and ABIDE datasets. The best result is highlighted in **bold**. For multiclass datasets of ADNI and PPMI, all these metrics are the same as accuracy in Table 6.1.

	model	precision	recall	micro-F1	ROC-AUC
Taowu	cs	76.67 ± 25.09	**81.67** ± 14.47	79.09	**77.50** ± 17.50
	ContrastPool	**78.33** ± 23.64	80.00 ± 25.82	**79.16**	**77.50** ± 17.50
Neurocon	SVM	68.33 ± 29.06	85.00 ± 32.02	75.76	65.00 ± 30.00
	ContrastPool	**75.83** ± 18.05	**86.67** ± 20.82	**79.13**	**68.33** ± 20.00
ABIDE	BrainNetCNN	63.79 ± 3.09	64.82 ± 4.90	64.30	66.68 ± 2.53
	ContrastPool	**64.48** ± 4.08	**66.10** ± 9.13	**65.28**	**68.16** ± 3.61

lowering the chance of rejecting the null hypothesis.

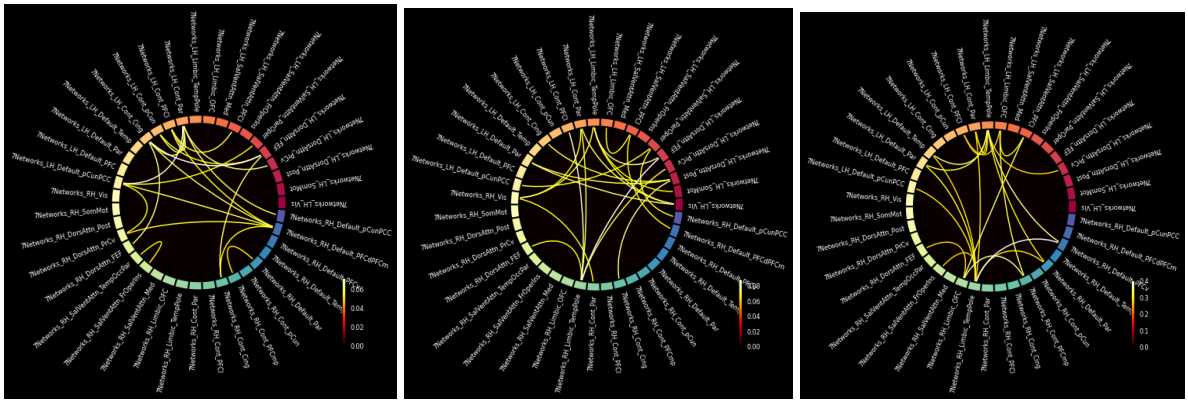
It is observed from Table 6.1 that different methods exhibit varied standard deviations on different datasets. Neural network models typically have a larger number of parameters than conventional ML methods, making them susceptible to overfitting, especially on small datasets like Taowu. Consequently, these methods typically exhibit higher standard deviations than conventional ML methods when tested on the same dataset. Regarding the comparison of standard deviation across different datasets, large datasets such as ABIDE are more likely to guarantee consistent data distribution across different folds, and thus tend to achieve smaller standard deviations than small datasets such as Taowu and Neurocon. Remarkably, the standard deviation obtained by our ContrastPool lies at the lower end among all with general GNNs and models for brain networks on all datasets. This also demonstrates that ContrastPool is able to remit the problem of overfitting. We further provide a hyperparameter analysis in Appendix C.2.

Apart from accuracy, we also report other evaluation metrics, including precision, recall, micro-F1, and ROC-AUC, of the top 2 models on each dataset. As shown in Table 6.2, ContrastPool performs the best on all datasets over all these metrics except for a single case (recall on Taowu when compared with cs). Note that we do not report the additional metrics on the two multi-class datasets PPMI and ADNI. This is because in the multi-class case, all these metrics are the same as accuracy, with the superiority of ContrastPool over all baselines already demonstrated in Table 6.1.



(a) ABIDE dataset. (b) ADNI dataset. (c) PPMI dataset.

Fig. 6.2: Contrast graph visualization. (a) Blue edges denote higher attention on TC group and red edges denote higher attention on ASD group. (b) Blue edges denote higher attention on CN/SMC group and red edges denote higher attention on AD/LMCI group. (c) Blue edges denote higher attention on NC group and red edges denote higher attention on PD group.



(a) On the ABIDE dataset. (b) On the ADNI dataset. (c) On the PPMI dataset.

Fig. 6.3: hord diagrams of contrast graphs. Only the edges with top-20 ROI-wise attention scores are shown for better visualization. (a) ROIs related to prefrontal cortex, parietal and cingulate are highlighted for Autism. (b) ROIs related to parietal and posterior are highlighted for Alzheimer's. (c) ROIs related to temporal and ventral prefrontal cortex are highlighted for Parkinson's.

6.3.3 Case Studies

In this subsection, we present case studies for ROI-wise attention, subject-wise attention, assignment matrix and generalization performance to showcase how ContrastPool meets the need of three characteristics of fMRI data.

Interpretation of ROI-wise Attention. To interpret the rationality of ContrastPool, we visualize the learnt contrast graphs on ABIDE, ADNI, and PPMI datasets. In order to visualize the ROI-wise attentions that are discriminative across different groups, we plot out the ROI-wise attention scores stored in the contrast graph. Essentially, each ROI attention in the contrast graph is obtained by averaging over all subjects (Eq. (10)) and contrasting between two groups (Eq.

(4)). As shown in Fig. 6.2, we select the edges with the top 10 ROI-wise attention weights. The chord diagram in Fig. 6.3 displays the edges with top-20 ROI-wise attention scores. We merge the ROIs in the same area in the chord diagram for a clearer visualization. Take the ABIDE dataset in Figs. 6.2(a) and 6.3(a) as an example. We can see that the contrast graph differentiates ROI pairs (edges in brain networks) with various importance levels (attention weights) and highlights a number of functional connections with high importance in distinguishing subjects from ASD and TC. Connections between the prefrontal cortex, parietal and cingulate are highlighted by our ROI-wise attention. Some ASD-specific neural machianism [181] may be underlying these connections, and these ROIs were regarded as key involved regions in previous ASD studies [182, 183]. Similar ROI-wise interpretations are found on Alzheimer's and Parkinson's as well. ROIs related to parietal and posterior are highlighted on ADNI (as shown in Figs. 6.2(b) and 6.3(b)), while those within temporal and ventral prefrontal cortex are highlighted on PPMI (as shown in Figs. 6.2(c) and 6.3(c)). These findings also match the domain knowledge in prior research of Alzheimer's [184–187] and Parkinson's [188–190].

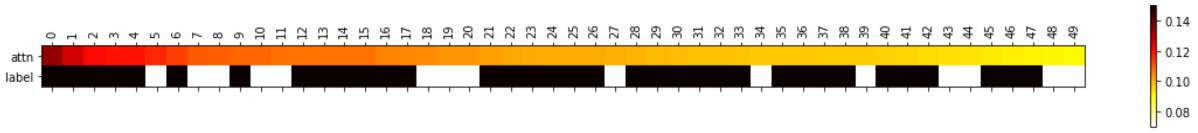


Fig. 6.4: The heatmap to top 50 subjects in the subject-wise attention on ADNI dataset. The black and white label denotes AD and MCI subjects, respectively.

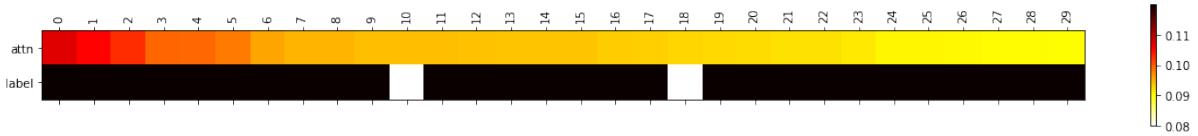


Fig. 6.5: The heatmap to top 30 subjects in the subject-wise attention on PPMI dataset. The black and white label denotes PD and SWEDD subjects, respectively.

Interpretation of Subject-wise Attention. We design an experiment on ADNI dataset to demonstrate ContrastPool's capability of underscoring informative subjects. Specifically, we merge the MCI group (82 subjects) with the AD group (143 subjects) into a single group (without revealing to ContrastPool the exact group each subject is from) to be used to contrast

against the CN group. Fig. 6.4 shows the subjects with the top 50 attention weights. The proportion of AD subjects in the top 50 attention subjects (35/50) is much higher than its proportion in the merged group of subjects (143/225). The top 5 highest attention subjects are all from the AD group. This observation illustrates that subject-wise attention could lead the model to focus more on typical/representative subjects in the dataset.

A similar conclusion can be drawn on PPMI (the other multi-class dataset) as well. We merge the SWEDD group (12 subjects) with the PD group (89 subjects) into a single group to be used to contrast against the NC group. Fig. 6.5 shows the subjects with the top 30 attention weights. The proportion of PD subjects in the top 30 attention subjects (28/30) is much higher than its proportion in the merged group of subjects (89/101). The top 10 highest attention subjects are all from the PD group, which once again demonstrates the ability of ContrastPool to automatically highlight representative subjects.

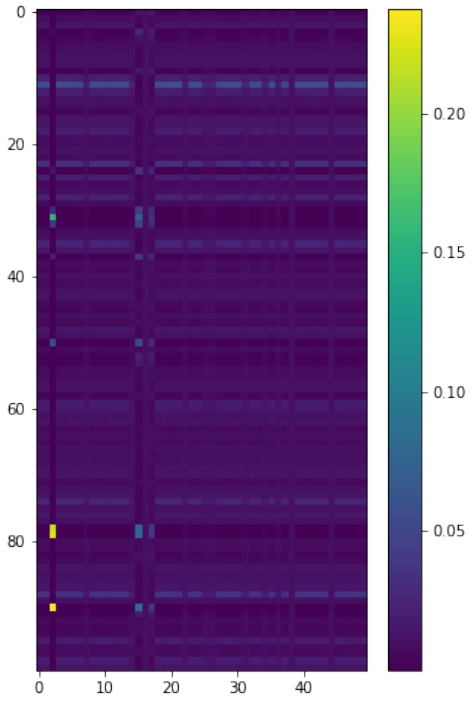


Fig. 6.6: Visualization of the assignment matrix $\boldsymbol{S}^{(1)}$ at the first layer of ContrastPool on PPMI.

Assignment Matrix. To better understand how the contrast graph works in the differentiable graph pooling, we provide a visualization of the assignment matrix $\boldsymbol{S}^{(1)}$ at the first layer of ContrastPool on the PPMI dataset in Fig. 6.6. Each row corresponds to an ROI and each

column is a cluster. We observe that the differentiable pooling is able to emphasize the most important ROIs and enforce the scores of most of the other ROIs to be close to 0. The entries with high values belong to temporal and ventral prefrontal cortex, which are consistent with the ROIs that have been highlighted in the contrast graph. This demonstrates that the differentiable pooling is well guided by the contrast graph in highlighting the discriminative ROIs.

Generalization Performance. We also observe that ContrastPool is able to alleviate the overfitting problem, which is a common problem when applying GNNs to brain networks. An example is shown in Fig. 6.7, where we plot the accuracy curves of ContrastPool and DiffPool on ABIDE dataset. The test accuracy of DiffPool decreases after its training accuracy coverages to 1. ContrastPool narrows dramatically the gap between the training and test accuracy, which demonstrates its ability in remitting the overfitting problem.

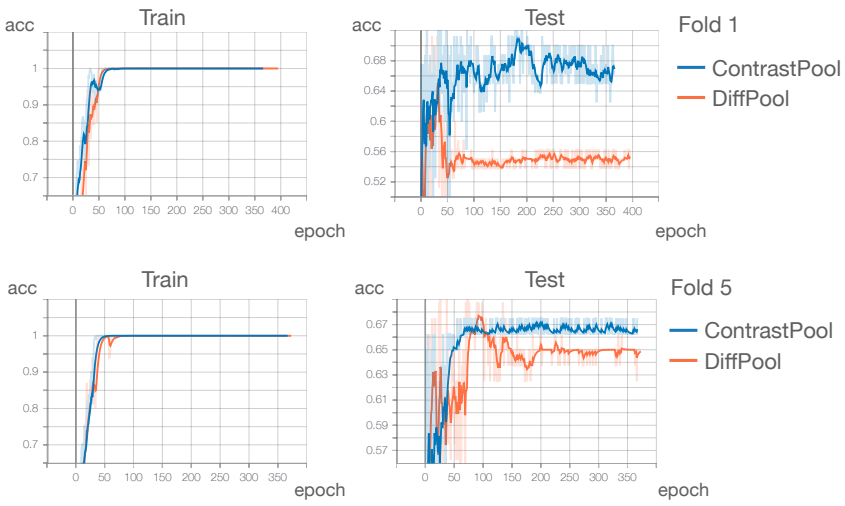


Fig. 6.7: Accuracy curves of ContrastPool and DiffPool on two folds on ABIDE dataset.

6.3.4 Ablation Studies

In this subsection, we validate empirically the design choices made in different components of our model: 1) the dual-attention mechanism and the contrast operation in the Contrastive Dual-Attention (CDA) block; and (2) the loss function. All experiments are conducted on the ABIDE dataset.

Contrastive Dual-attention Block. Our CDA block performs dual attention (subject-wise and ROI-wise) on both adjacency matrices and feature matrices. We disable each of them to inspect

their effects. The results reported in Table 6.3 demonstrate that the CDA with all attentions enabled achieves the best performance. In particular, if the dual attention is entirely disabled (row 1), the summary graph is essentially obtained by averaging over all graphs in each group, which compromises the performance. Moreover, relying solely on ROI-wise attention (row 2) may lead to a further deterioration in performance. This implies that without the support of subject-wise attention, the ROI-wise attention suffers from the information extracted from less representative subjects.

To verify the effectiveness of the contrast operation, we conduct an experiment with a CDA variant without considering groups. The results (the last two rows of Table VII) show that ContrastPool with the contrast operation obtains significantly better performance. This verifies the necessity of capturing the discrepancy across groups in the model design.

Table 6.3: Ablation Study on CDA block on ABIDE. Winner is highlighted in **bold**.

Adjacency Matrices		**Feature Matrices**		**Contrast**	**acc ± std**
subject-wise	ROI-wise	subject-wise	ROI-wise		
				✓	65.88 ± 4.11
	✓		✓	✓	64.63 ± 3.83
✓		✓		✓	66.25 ± 2.68
		✓	✓	✓	66.88 ± 4.23
✓	✓			✓	67.88 ± 4.78
✓	✓	✓	✓		62.71 ± 3.08
✓	✓	✓	✓	✓	**68.63** ± 2.65

Loss Function. We test our design of the loss function by disabling two entropy losses. As shown in Table 6.4, the results demonstrate that both $\mathcal{L}_{E_1}$ and $\mathcal{L}_{E_2}$ are effective in boosting the model performance.

Table 6.4: Ablation Study on Entropy Loss on ABIDE. The best result is highlighted in **bold**.

$\mathcal{L}_{cls}$	$\mathcal{L}_{E_1}$	$\mathcal{L}_{E_2}$	**acc ± std**
✓			64.88 ± 4.45
✓	✓		66.13 ± 2.65
✓	✓	✓	**68.63** ± 2.65

6.4 Summary

This section proposes a novel GNN-based solution for brain network classification, taking the unique characteristics of the underlying fMRI data into account. Our proposed method, *ContrastPool*, can adaptively select the most discriminative regions of interest and the most representative subjects by engaging a contrastive dual-attention block. It allows for a flexible local information aggregation within each group. We demonstrate the superiority of our method over 17 state-of-the-art baselines on 5 brain-network datasets spanning over 3 diseases. Moreover, our case studies show the interestingness, simplicity, and high explainability of the patterns extracted by our method, which find consistency in the neuroscience literature. We hope our work can inspire further research in leveraging GNNs for brain network analysis, and show significance in real-world applications, such as the early diagnosis and personalized treatments of neurodegenerative diseases.

Chapter 7

Contrastive Transformer

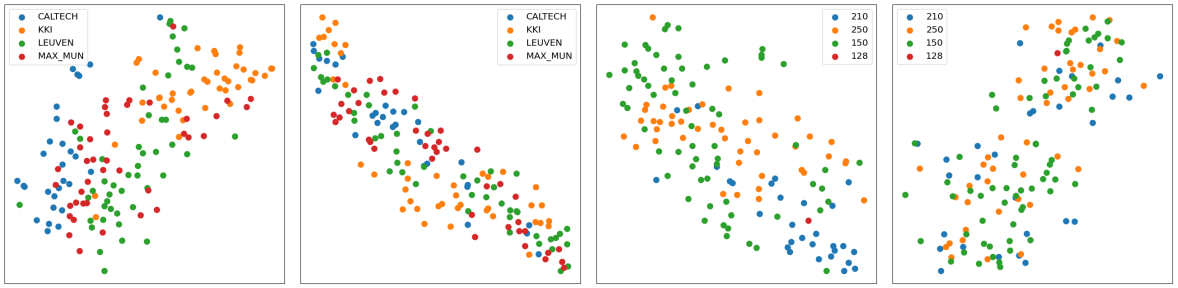
While recent efforts in the realm of GNNs have introduced specialized designs for brain networks and addressed issues related to node identity [24, 25, 81], many of them overlook capturing group-specific patterns. Neglecting this aspect can lead to models that overfit outliers and hinder interpretability [117]. One notable advancement, ContrastPool [179], introduces a dual attention mechanism to extract discriminative features across ROIs for subjects within the same group. However, it is hard to optimize the contrast graph via classification loss and the high computational complexity limits its further application.

In this chapter[1], we present *Contrasformer*, a novel contrastive brain network Transformer, that harnesses the distinctive properties of brain network data to fully leverage the capabilities of Transformer-based models for brain network analysis. Inspired by ContrastPool, this chapter employs a dual attention block to create a contrast graph that encodes group-specific information. Instead of graph pooling, our approach integrates the contrast graph with the encoded brain network using an attention mechanism. In contrast to previous work, this chapter further takes advantage of node identity awareness by introducing a contrastive loss to constrain that identical ROIs across subjects have similar representations. Additionally, a cluster loss is introduced to guarantee group consistency in graph representations.

[1]The work in this chapter has been published as "Contrasformer: A Brain Network Contrastive Transformer for Neurodegenerative Condition Identification", in 33rd ACM International Conference on Information and Knowledge Management, 2024.

7.1 Motivation

While graph neural networks (GNNs) [43, 152, 153] and graph Transformers [72, 75, 76] have recently been adopted in a wide range of graph-related tasks, applying them to brain networks faces two challenges below in capturing the disease-specific pattern.



(a) Site distribution for original feature.

(b) Site distribution for Contrasformer representation.

(c) Scanning duriation distribution for original feature.

(d) Scanning duriation distribution for Contrasformer representation.

Fig. 7.1: The distribution of the original feature and Contrasformer representation for subjects from multiple sites and scanning duriation in ABIDE dataset. Each point in the figure represents a subject and different colors denote the sites these subjects are acquired from or their lengths of the BOLD signals. The representation of each subject is obtained by mean pooling and visualized by t-SNE [1]. Compared with (b) and (d), (a) and (c) exhibit obvious distribution shifts.

Sub-population-specific (SPS) Noise. Analyzing neurological disorders aims to capture disease-specific patterns that are invariant across all populations. However, certain features are common to a sub-population (not generalizable to the entire population) and are unrelated to the disease, constituting sub-population-specific (SPS) noise. For instance, brain network datasets like Autism Brain Imaging Data Exchange (ABIDE) [34] and Alzheimer's Disease Neuroimaging Initiative (ADNI) [122] are collected from multiple sites. Subjects from different sites may exhibit site differences (scanner variability, different inclusion/exclusion criteria) [191]. Such noise could lead the model to focus on the site-specific pattern instead of learning population-invariant information. Besides, the varying scanning duration could result in different periods of region activation recorded for subjects. Furthermore, label inconsistencies may arise due to differences in diagnostic criteria used by doctors labeling these subjects. Fig. 7.1(a) provides an example of the site distribution of subjects in the ABIDE dataset. Each point in the figure represents a subject, and different colors denote the sites from which these subjects are

acquired. From this example, it can be observed that subjects from the same site tend to have similar features, as the site-specific noise dominates the similarity. A similar observation is found in the distribution of sub-populations with different scanning durations. As illustrated in Fig. 7.1(c), each point represents a subject, and different colors denote the lengths of the BOLD signal for these subjects. Such a conspicuous distribution shift across sub-populations could easily mislead the model to overfit the SPS noise, thereby limiting its performance.

Node-identity awareness. The construction of brain networks requires a specific parcellation method to split the whole brain into several ROIs. The same parcellation method is applied to all subjects, and thus the ROI definition is identical across all brain networks. Such a property does not generally exist in other graph-structured data, necessitating our specialized tailoring for brain networks. Existing general-purposed GNNs are designed to learn the structural pattern of graphs without considering their node identities [43, 105, 192].

While some recent GNNs have introduced specialized designs for brain networks and addressed issues related to node identity [24, 25, 81], many of them overlook addressing SPS noise and capturing group-specific patterns. Neglecting these aspects can lead to models that overfit outliers and hinder interpretability [117].

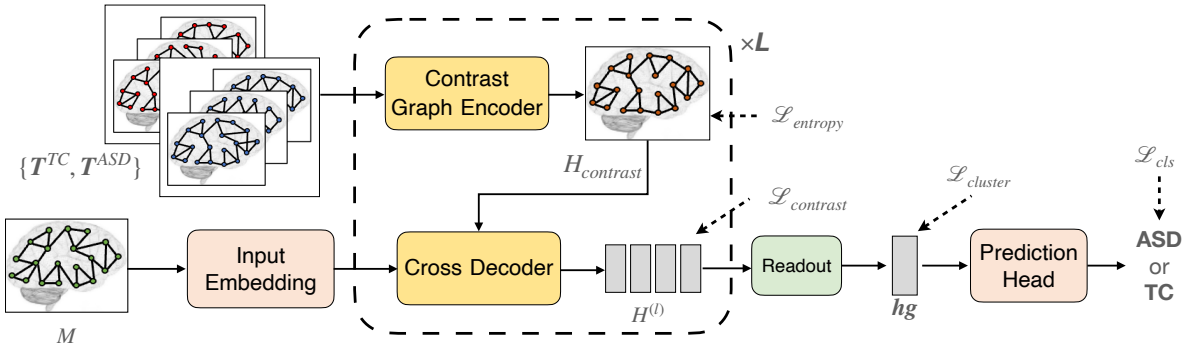


Fig. 7.2: The architecture of Contrasformer, using Autism as an example.

7.2 Methodology

In this section, we provide a detailed exposition of the design of our proposed Contrasformer, depicted in Fig. 7.2. Contrasformer adopts an encoder-decoder architecture, featuring three key components: (1) A contrast graph encoder is introduced (in Section 7.2.1) to extract the most

discriminative task-related features from the training set. To alleviate the SPS noise, a two-stream attention mechanism is employed to generate a contrast graph that captures the invariant information across all sub-population. The learnt contrast graph is then utilized in both training and test stages to be incorporated with the brain network representation learning. (2) A cross decoder is introduced (in Section 7.2.2) to combine the input brain network with node identity, and subsequently fuse the contrast graph with the identity-embedded brain network by a cross-attention to update node representations. (3) A classification loss $\mathcal{L}_{cls}$ and three auxiliary losses $\mathcal{L}_{entropy}$, $\mathcal{L}_{cluster}$, $\mathcal{L}_{contrast}$ are incorporated (in Section 7.2.3) to guide the end-to-end training. These losses emphasize the node identity of ROIs and consider group-level relationships.

7.2.1 Contrast Graph Encoder with Two-stream Attention

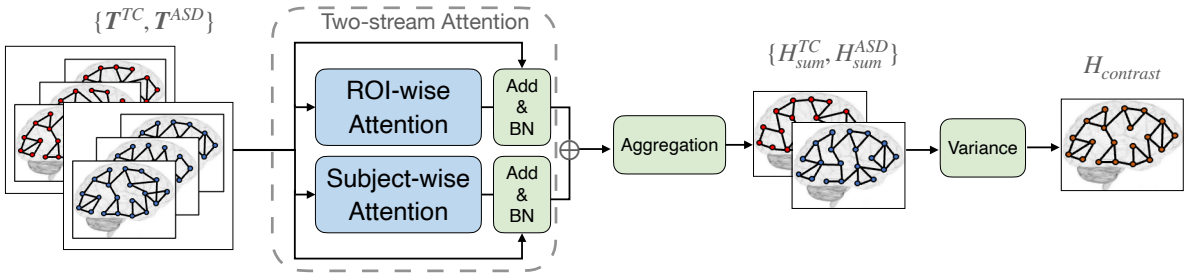


Fig. 7.3: The architecture of contrast graph encoder. Each group of brain networks is fed into the two-stream attention to obtain a summary graph. The contrast graph is generated by contrasting the summary graphs of different groups.

To generate a contrast graph with group-specific information, we introduce a two-stream contrast graph encoder. In neuroscience, normally there are only some ROIs in the brain that can reflect the lesion of a neurological disorder. So the aim of this encoder is to extract the most discriminative ROIs for each group while adaptly learning the contribution of each subject. Such a two-stream attention design is able to capture the population-invariant information embedded in subjects and ROIs, which alleviates the impact of SPS noise in the downstream task.

Taking Autism as an example, we use tensors $\boldsymbol{T}^{TC} \in \mathbb{R}^{n^{TC} \times m \times m}$ and $\boldsymbol{T}^{ASD} \in \mathbb{R}^{n^{ASD} \times m \times m}$ to denote all brain networks of TC and ASD groups in the training set, respectively. n^{TC} and n^{ASD} denote the number of subjects in TC/ASD groups and m denotes the number of ROIs. As

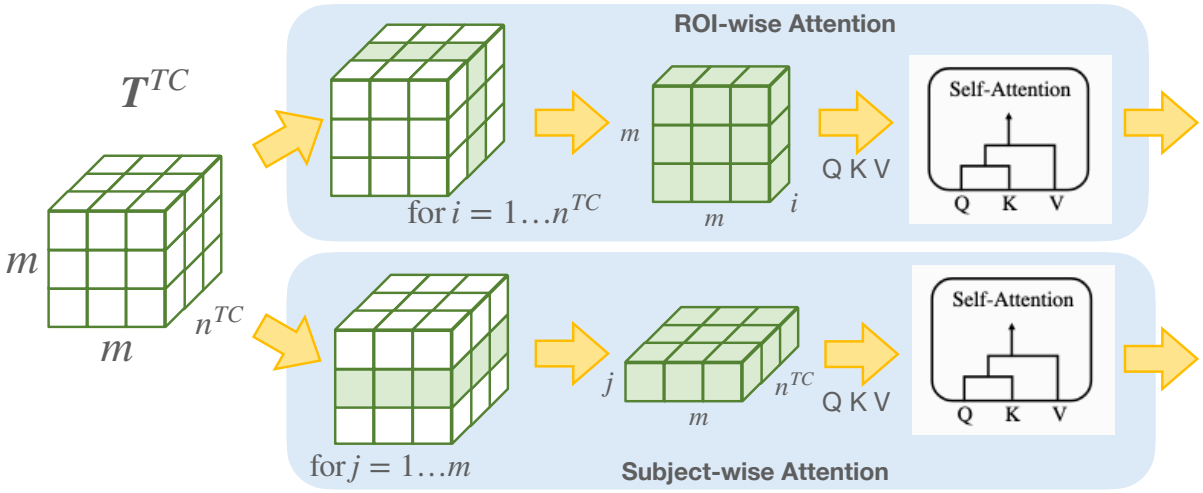


Fig. 7.4: The detail of two-stream attention using TC as an example. The ROI- and subject-wise attention blocks compute self-attention from different views of the input. Parameters of self-attention inside these two branches are independent.

illustrated in Fig. 7.3, given the input set of brain networks that belong to different groups, the objective of the two-stream attention is to generate a summary graph for each group. An ROI-wise attention and a subject-wise attention are first computed independently for each group of brain networks. We adopt the self-attention mechanism [72] for the ROI- and subject-wise attention. In general, given a matrix $\boldsymbol{X} \in \mathbb{R}^{k \times d}$, where k and d are arbitrary integers, the self-attention function can be written as:

$$\text{Attn}(\boldsymbol{X}) = \text{norm}\left(\boldsymbol{X} + \phi\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathsf{T}}}{\sqrt{k}}\right)\boldsymbol{V}\right), \tag{7.1}$$

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K, \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V, \tag{7.2}$$

$$\phi(\boldsymbol{z})_i = \text{softmax}(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{d} e^{z_j}}, \text{for } i = 1 \ldots k, \boldsymbol{z} \in \mathbb{R}^k, \tag{7.3}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d \times d}$ are parameter matrices, e is the Euler's number, and norm$(\cdot)$ is a normalization function.

For the brain networks of TC group $\boldsymbol{T}^{TC}$, the detail of two-stream attention is shown in Fig. 7.4. In ROI-wise attention $\text{Attn}_{ROI}(\cdot)$, for each subject $i \in \{1 \ldots n^{TC}\}$, we compute the self-attention across all ROIs in this subject. The most informative ROIs inside each subject will be extracted in this way. Similarly, in subject-wise attention $\text{Attn}_{subject}(\cdot)$, for each ROI

$j = 1 \dots m$, we compute the self-attention across all subjects in this group. The input of subject-wise attention is implemented by simply transposing the subject and ROI dimensions. This operation helps our model highlight the most discriminative subjects of a certain ROI. The rationale of such attention mechanism is that we intend to extract the group-invariant feature and filter out the SPS noise.

Note that the normalization function we are using here (in Fig. 7.3) is batch normalization (BN) instead of the commonly used layer normalization (LN) in Transformer [72]. It is because (1) the input of the two-stream attention is constant for each training step; (2) the lengths of input sequences are consistent (m for ROI-wise attention, n^{TC} for subject-wise attention); (3) we aim to highlight the consistency within each group. We conduct an ablation study in Section 7.3.3 to verify the effectiveness of our model architecture.

Afterward, the summary graph of the TC group $\boldsymbol{H}^{TC}_{sum} \in \mathbb{R}^{m \times m}$ is generated as:

$$\boldsymbol{H}^{TC}_{sum} = \frac{1}{n^{TC}} \sum_{i=1}^{n^{TC}} (\boldsymbol{H}^{TC}_{ROI} + \boldsymbol{H}^{TC}_{subject})(i, :, :), \tag{7.4}$$

$$\boldsymbol{H}^{TC}_{ROI} = \left[\text{Attn}_{ROI} \left(\boldsymbol{T}^{TC}(i, :, :) \right) : i = 1, \dots, n^{TC} \right], \tag{7.5}$$

$$\boldsymbol{H}^{TC}_{subject} = \left[\text{Attn}_{subject} \left(\boldsymbol{T}^{TC}(:, j, :) \right) : j = 1, \dots, m \right], \tag{7.6}$$

where $\boldsymbol{H}^{TC}_{ROI}, \boldsymbol{H}^{TC}_{subject} \in \mathbb{R}^{n^{TC} \times m \times m}$ denote two 3-dimensional matrices to store the outputs of the ROI-wise and subject-wise attention functions, and $[\cdot]$ denotes the stack function to combine a set of matrices to a single higher-dimensional matrix. The summary graphs of other groups can also be obtained in a similar way. Once we obtain the summary graphs $\mathcal{H}_{sum} = \{\boldsymbol{H}^{TC}_{sum}, \boldsymbol{H}^{ASD}_{sum}\}$ for TC and ASD groups, the contrast graph is generated by computing the variance of all summary graphs:

$$\boldsymbol{H}_{contrast} = \frac{1}{|\mathcal{H}_{sum}|} \sum_{\boldsymbol{H}_{sum} \in \mathcal{H}_{sum}} \left(\boldsymbol{H}_{sum} - \bar{\boldsymbol{H}}_{sum} \right)^2, \tag{7.7}$$

$$\bar{\boldsymbol{H}}_{sum} = \frac{1}{|\mathcal{H}_{sum}|} \sum_{\boldsymbol{H}_{sum} \in \mathcal{H}_{sum}} \boldsymbol{H}_{sum}. \tag{7.8}$$

The generated contrast graph $\boldsymbol{H}_{contrast}$ contains the discriminative information about the specific disease, which can be incorporated with the cross decoder to boost the downstream brain network representation learning. The contrast graph generation also works for datasets with multiple groups by making contrast among all groups. Note that we only use subjects in the training set to train the contrast graph encoder. The generated contrast graph is used in the test stage as prior knowledge to avoid data leakage.

7.2.2 Cross Decoder with Identity Embedding

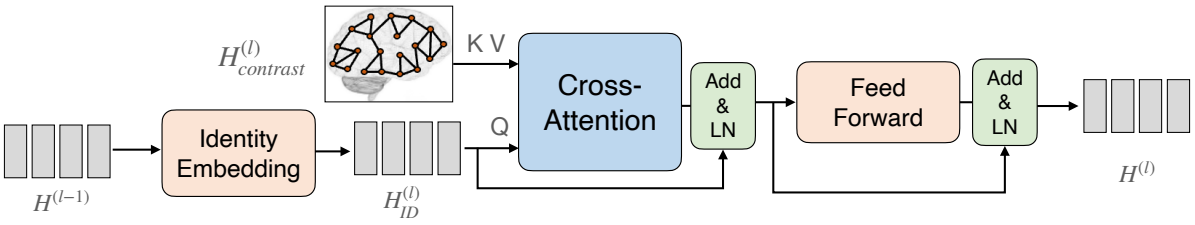


Fig. 7.5: The architecture of the cross decoder. The generated contrast graph is incorporated with the identity-embedded brain network by a cross-attention for the downstream representation learning.

By leveraging the group-discriminative information captured in the contrast graph, the cross decoder of Contrasformer aims to produce a high-quality representation for each input brain network (Fig. 7.5).

In graph transformer models, positional embedding is commonly used to encode the topological information of the graph. However, existing designs for general graph representation learning, such as distance-based, centrality-based and eigenvector-based positional embedding [75, 193, 194], can hardly migrate to brain network due to its high density (always fully connected). The correlation-based brain networks naturally contain sufficient positional information for ROIs. Therefore the general-purposed positional embedding is not only expensive but also redundant in our case.

Instead of positional embedding that captures the topological information of the graph structure, we propose a learnable identity embedding to adaptively learn the unique identity for each ROI. Such embedding attaches the same identity for nodes that belong to the same ROI. As shown in Eq. (7.9), we introduce a parameter matrix $\boldsymbol{W}^{(l)}_{ID} \in \mathbb{R}^{m\times m}$ to encode the identity of nodes. $\delta(\cdot)$ denotes a multilayer perceptron (MLP).

$$\boldsymbol{H}^{(l)}_{ID} = \boldsymbol{H}^{(l-1)} + \delta(\boldsymbol{H}^{(l-1)} + \boldsymbol{W}^{(l-1)}_{ID}). \tag{7.9}$$

After identity embedding, we combine the contrast graph with the encoded brain network by a cross-attention function followed by a layer normalization. The encoded brain network $\boldsymbol{H}^{(l)}_{ID}$ serves as $\boldsymbol{Q}$ while the contrast graph $\boldsymbol{H}^{(l)}_{contrast}$ is treated as $\boldsymbol{K}$ and $\boldsymbol{V}$ in Eq. (7.1). Each input brain network is fed into the cross-attention module to query the task-specific information in

the contrast graph. The intuition here is to use the contrast graph as prior knowledge to guide the brain network representation learning. By hiding the non-indicative ROIs/connections and emphasizing the indicative ones, task-specific domain knowledge is introduced to the embedded brain networks.

In addition to the cross-attention sub-layer, a position-wise feed-forward network (FFN) with a layer normalization function is applied to each position to get the output node representations $\boldsymbol{H}^{(l)}$ of the l-th Contrasformer layer. The FFN is applied to each position separately and identically [72]. After L layers of Contrasformer, a readout function Readout$(\cdot)$ is applied to the node representations to generate a graph representation: $\boldsymbol{hg}$ = Readout$(\boldsymbol{H}^{(l)})$. The graph representation is then passed to the prediction head for classification.

7.2.3 Loss Functions

In order to introduce domain knowledge and make model optimization easier to converge, we utilize 4 loss functions to guide the end-to-end training. (1) A commonly-used cross-entropy loss $\mathcal{L}_{cls}$ [167] for graph classification; (2) an entropy loss $\mathcal{L}_{entropy}$ for contrast graph sparsification; (3) a cluster loss $\mathcal{L}_{cluster}$ to take the group relationship (i.e., similarity and discrepancy) into account; (4) a contrastive loss $\mathcal{L}_{contrast}$ to constrain node-identity awareness. The total loss is computed by:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \lambda_1 * \mathcal{L}_{entropy} + \lambda_2 * \mathcal{L}_{cluster} + \lambda_3 * \mathcal{L}_{contrast}, \tag{7.10}$$

where λ_1, λ_2 and λ_3 are trade-off hyperparameters for balancing different losses.

Entropy Loss. To prevent a smooth contrast graph that treats all ROIs equally, risking the loss of discriminative ability, we introduce a sparsity constraint. To achieve this, we employ an entropy loss, compelling the model to prioritize the most task-specific ROI connections. The entropy loss is formulated as follows:

$$\mathcal{L}_{entropy} = \frac{1}{m} \sum_{i=1}^{m} \text{entropy}\left(\boldsymbol{H}_{contrast}(i,:)\right), \tag{7.11}$$

$$\text{entropy}(\boldsymbol{p}) = -\sum_{j=1}^{m} \boldsymbol{p}_j \log(\boldsymbol{p}_j). \tag{7.12}$$

Cluster Loss. Most existing GNN/Transformer architectures treat individual input graphs independently during training. Neglecting the relationships between classes could lead to a significant compromise in model effectiveness for downstream classification tasks [158]. For our application, we want to find the common patterns/biomarkers for a certain neurological disorder identification task. Thus we propose a cluster loss to leverage graph-level similarity and make the graph representations more sepratable:

$$\mathcal{L}_{cluster} = \log \frac{\exp(\sum_{c \in C} \sigma_c^2)}{\exp(\sum_{c \in C} \sum_{i \in C} \|\mu_c - \mu_i\|_2)}, \tag{7.13}$$

$$\mu_c = \sum_{k \in \mathcal{S}^c} \frac{\boldsymbol{hg}^k}{|\mathcal{S}^c|}, \quad \sigma_c^2 = \sum_{k \in \mathcal{S}^c} \frac{(\boldsymbol{hg}^k - \mu_c)^2}{|\mathcal{S}^c|}, \tag{7.14}$$

where μ_c and σ_c denote the mean and standard deviation of graph representations belonging to group c, $\mathcal{S}^c$ denotes the subject indices that belong to group c, C denotes the set of classes, and $\boldsymbol{hg}^k$ denotes the graph representation with index k in $\mathcal{S}^c$. As shown in Fig. 7.6, the cluster loss aims to pull the graph representations within a group close to each other and push the centers of groups as far as possible. By using such cluster loss, the group-level relationship of all classes is considered equally no matter how many subjects it contains.

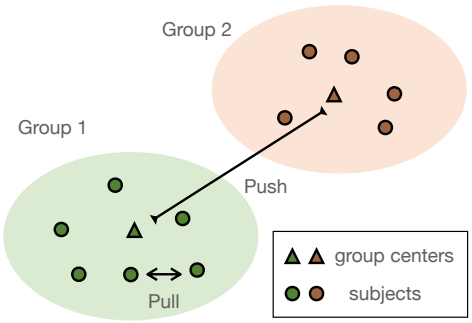


Fig. 7.6: The cluster loss enforces subjects that belong to the same group to get similar representations, the subjects from different groups less similar.

Contrastive Loss. Existing graph contrastive learning technologies [55, 195] require data argumentation by modifying graph structure or dropping node/edge features. Such contrast is still limited to each individual graph. It also cannot be migrated to our brain networks because the connectivity matrix naturally contains its structural and positional information.

Herein, we design an ROI-level contrastive loss to further leverage the node identity of ROIs. Thanks to the unique property of brain networks' node-identity awareness, we are able to conduct contrastive learning by aligning ROIs across subjects. To the best of our knowledge, this is the first attempt to bring in contrastive constraints at the ROI level for brain network analysis. The contrastive loss $\mathcal{L}_{contrast}$ shown in Eq. (7.15) is defined to enforce maximizing the consistency between positive pairs compared with negative pairs [54]. We use $\boldsymbol{H}_j^i$ to denote the node representation for the j-th node in the i-th subject, where $j = 1, \ldots, m$, $i = 1, \ldots, n$, and n is the total number of subjects in the training set. We denote the set of positive pairs as $\mathcal{P}^{pos} = \{(\boldsymbol{H}_j^i, \boldsymbol{H}_j^p)\}$ and the set of negative pairs as $\mathcal{P}^{neg} = \{(\boldsymbol{H}_j^i, \boldsymbol{H}_r^q)\}$, where $p = 1, \ldots, n, p \neq i$, $q = 1, \ldots, n$, $r = 1, \ldots, m, r \neq j$. A temperature hyper-parameter τ [55] is introduced to control the smoothness of the probability distribution, and $\text{sim}(\cdot)$ denotes the cosine similarity function. As elaborated in Fig. 7.7, we treat the same ROI of all subjects as positive pairs, and different ROIs from the same/different subjects as negative pairs to emphasize the node-identity awareness of brain networks.

$$\mathcal{L}_{contrast} = -\log \frac{\sum_{(\boldsymbol{H}_j^i, \boldsymbol{H}_j^p) \in \mathcal{P}^{pos}} \exp(\text{sim}(\boldsymbol{H}_j^i, \boldsymbol{H}_j^p)/\tau)}{\sum_{(\boldsymbol{H}_j^i, \boldsymbol{H}_r^q) \in \mathcal{P}^{neg}} \exp(\text{sim}(\boldsymbol{H}_j^i, \boldsymbol{H}_r^q)/\tau)}. \tag{7.15}$$

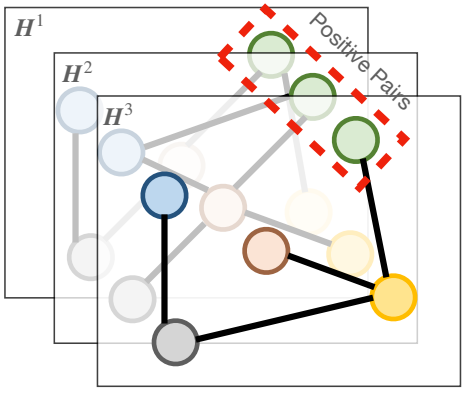


Fig. 7.7: The contrastive loss treats the nodes belonging to the same ROI as positive pairs, and all the other node pairs are considered negative pairs.

7.3 Experimental Study

In this section, we assess the performance of Contrasformer in comparison with 13 baseline models. We also present two case studies to provide the domain interpretation of the generated contrast graph and discuss the generalization ability of our method. We further conduct ablation studies to analyze the effects of the components in Contrasformer. In the end, we report the time efficiency of both our method and baseline models. We remit the introduction of the brain network datasets, the baseline models and the implementation details since they are already described in the previous chapters.

7.3.1 Main Results

Table 7.1: Graph Classification Results (Average Accuracy ± Standard Deviation) over 10-fold-CV. The best result is highlighted in **bold**. The second-best result is underlined.

	Model	Mātai	PPMI	ADNI	ABIDE
Conventional ML methods	LR	60.00 ± 20.00	56.48 ± 6.76	61.97 ± 4.24	64.81 ± 3.70
	SVM	56.67 ± 17.00	63.21 ± 8.62	61.52 ± 4.95	64.41 ± 5.09
General-purposed GNNs	GCN	56.67 ± 18.56	54.02 ± 9.06	60.40 ± 4.89	60.19 ± 2.96
	GraphSAGE	61.67 ± 10.67	55.00 ± 12.89	59.35 ± 3.39	61.75 ± 4.35
	GAT	66.67 ± 18.33	54.98 ± 8.03	59.73 ± 2.85	60.10 ± 4.13
	GatedGCN	58.33 ± 8.33	52.60 ± 11.51	64.55 ± 1.87	61.66 ± 3.36
	GPS	63.33 ± 14.53	57.50 ± 7.83	65.17 ± 3.50	63.04 ± 3.36
Neural networks tailored for brain networks	BrainNetCNN	61.67 ± 13.33	57.33 ± 10.32	60.48 ± 3.29	65.75 ± 3.24
	LiNet	51.67 ± 13.84	60.71 ± 10.61	61.91 ± 1.02	54.05 ± 4.50
	PRGNN	55.00 ± 16.75	58.83 ± 6.89	62.51 ± 3.36	59.71 ± 4.54
	BrainGNN	53.33 ± 24.49	61.71 ± 6.05	61.05 ± 1.23	62.88 ± 2.46
	BNTF	61.67 ± 11.17	51.60 ± 6.15	65.49 ± 3.25	63.70 ± 4.84
	ContrastPool	61.67 ± 13.02	64.00 ± 6.63	65.67 ± 6.64	65.01 ± 3.84
Ours	Contrasformer	**68.33** ± 18.93	**67.00** ± 4.58	**69.33** ± 3.63	**66.27** ± 3.67

We report the accuracy on 4 brain network datasets in Table 7.1. Our proposed Contrasformer consistently outperforms all 13 baselines on all datasets. In particular, Contrasformer improves over all networks specifically designed for brain networks on these four datasets by up to 10.8% ((68.33%-61.67%)/61.67% = 10.8% on Mātai). These experimental results demonstrate the effectiveness of our brain network oriented model design. The improvement may result from two reasons. First, the participation of the contrast graph in brain network representation learning provides reasonable and discriminative information about certain conditions.

Table 7.2: Results of more evaluation metrics on ABIDE. The best result is highlighted in **bold**. The second-best result is underlined.

	Model	Precision	Recall	micro-F1	ROC-AUC
General-purposed GNNs	GCN	59.69 ± 5.50	57.56 ± 7.01	58.49 ± 5.78	61.08 ± 4.92
	GraphSAGE	60.33 ± 4.91	58.20 ± 7.85	58.99 ± 5.45	61.59 ± 4.36
	GAT	59.57 ± 4.63	55.11 ± 7.89	56.96 ± 5.16	60.43 ± 3.88
	GatedGCN	61.65 ± 4.12	55.74 ± 11.58	58.05 ± 8.20	62.31 ± 4.32
	GPS	59.97 ± 5.36	68.75 ± 11.22	63.48 ± 5.98	63.34 ± 5.15
Neural networks tailored for brain networks	BrainNetCNN	**63.98** ± 3.47	61.25 ± 5.66	62.39 ± 3.13	64.78 ± 2.52
	LiNet	56.34 ± 6.94	30.37 ± 9.23	38.66 ± 8.32	54.39 ± 3.29
	PRGNN	60.83 ± 7.44	53.68 ± 8.36	56.77 ± 7.15	61.01 ± 5.74
	BrainGNN	62.48 ± 5.92	57.15 ± 4.66	59.42 ± 3.23	62.74 ± 3.58
	BNTF	60.34 ± 5.40	70.19 ± 8.66	64.64 ± 5.65	64.15 ± 5.42
	ContrastPool	63.56 ± 3.62	61.45 ± 5.43	62.28 ± 2.81	64.52 ± 2.30
Ours	Contrasformer	62.59 ± 4.40	**73.07** ± 4.69	**67.25** ± 3.01	**66.62** ± 3.47

Second, the properties of fMRI and group constraints are introduced to the model training by dedicated loss functions.

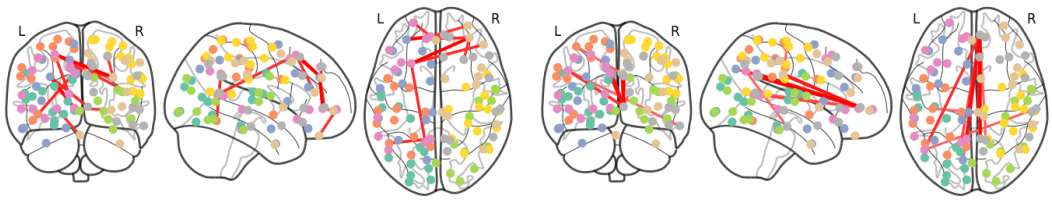
Apart from accuracy, we also report other evaluation metrics, including precision, recall, micro-F1, and ROC-AUC, of all the models on the ABIDE dataset. As shown in Table 7.2, Contrasformer performs the best over all these metrics except for precision.

We can also discover that compared with other baselines, our Contrasformer can dramatically improve recall without sacrificing precision. Besides, in medical diagnostics, it's crucial to ensure that all individuals with a certain condition are correctly identified, even if it means some false positives. Missing a true positive (failing to diagnose a disease) can have severe consequences, while false positives can be further examined or retested. Therefore, models with higher recall rates, like our Contrasformer, are more suitable for application in real-life medical auxiliary diagnosis.

We provide the hyperparameter analysis in Appendix D.1.

7.3.2 Model Interpretation

In this subsection, we delve into the interpretability of our model by examining specific cases, including the visualization of the learnt contrast graph and an analysis of the generalization ability.



(a) ADNI dataset. (b) ABIDE dataset.

Fig. 7.8: Contrast graph visualization by highlighting the top 10 edges with the highest strength.

Contrast Graph Visualization. Despite the high accuracy achieved by our model, a critical concern is the interpretability of their decision-making process. In the context of brain biomarker detection, identifying salient ROIs associated with predictions as candidate/potential biomarkers is crucial. In this study, we leverage built-in model interpretability to explore disease-specific biomarker analysis. To interpret Contrasformer's reasoning, we visualize the learnt contrast graphs for Alzheimer's Disease and Autism on the ADNI and ABIDE datasets, by using the Nilearn toolbox [145]. We select edges with the top 10 attention weights. As depicted in Fig. 7.8(a), highlighted connections between the lateral prefrontal cortex, prefrontal cortex, and dorsal prefrontal cortex medial prefrontal cortex in the ADNI dataset suggest potential AD-specific neural mechanisms [196]. These regions have been recognized as key areas in previous AD studies [197, 198]. Similar ROI-wise interpretations are found in Autism. In Fig. 7.8(b), highlighted ROIs related to precuneus posterior cingulate cortex, cingulate, and dorsal prefrontal cortex medial prefrontal cortex on ABIDE align with domain knowledge from prior Autism research [182, 183, 199].

Generalization Ability. While task-specific biomarkers are valuable for identifying disease-relevant features, it is crucial to determine whether these biomarkers are invariant over the entire population, i.e., whether they generalize well across sub-populations and other diverse populations. To assess the generalization ability of Contrasformer, we conduct evaluations on subjects from previously unseen sites. Specifically, we designate two sites from the ABIDE dataset as the test set, while the remaining subjects are split into training and validation sets, maintaining an 8:1:1 ratio. The test set remains constant across 10 experiments with different train-validation splits, and the average results are reported in Table 7.3. Notably, Con-

Table 7.3: Results on ABIDE dataset when generalizing to unseen sites. The best result is highlighted in **bold**. The second-best result is underlined.

	Accuracy	Precision	Recall	micro-F1	ROC-AUC
GCN	55.15 ± 3.63	48.82 ± 3.54	56.67 ± 7.04	52.33 ± 4.69	55.32 ± 3.76
GatedGCN	59.03 ± 1.55	52.87 ± 1.67	55.33 ± 8.52	53.79 ± 4.96	58.61 ± 2.17
BrainNetCNN	61.55 ± 2.31	54.44 ± 2.10	74.44 ± 5.19	62.81 ± 2.45	63.00 ± 2.28
LiNet	53.50 ± 2.89	45.99 ± 4.07	35.11 ± 3.69	39.71 ± 3.26	51.43 ± 2.70
PRGNN	55.63 ± 2.34	49.21 ± 2.77	49.56 ± 3.86	49.35 ± 3.08	54.95 ± 2.43
ContrastPool	56.18 ± 2.01	48.60 ± 3.91	42.27 ± 10.77	44.61 ± 8.18	54.50 ± 2.61
Contrasformer (ours)	**64.61** ± 1.83	**56.52** ± 1.56	**77.95** ± 3.53	**65.50** ± 1.93	**66.22** ± 1.88

trasformer consistently outperforms all baseline models, demonstrating its robustness against SPS noise. The baseline models exhibit a significant performance reduction compared to Table 7.1, indicating the detrimental impact of SPS noise on their generalization ability. In contrast, Contrasformer, with its two-stream attention mechanism, effectively extracts and emphasizes task-related features, mitigating the adverse effects of SPS noise.

7.3.3 Ablation Study

In this subsection, we empirically validate the design of our model, including (1) the important modules; and (2) the loss functions. All experiments in this subsection are conducted on ABIDE dataset.

Table 7.4: Ablation study on important modules in Contrasformer on ABIDE dataset. The best result is highlighted in **bold**.

Attn_{ROI}	$\text{Attn}_{subject}$	batch norm	ID enc	Accuracy	Precision	Recall	micro-F1	ROC-AUC
	✓	✓	✓	63.14 ± 4.71	59.18 ± 5.36	75.76 ± 7.83	66.04 ± 4.11	63.77 ± 4.78
✓		✓	✓	63.63 ± 4.28	62.42 ± 6.02	60.00 ± 6.37	60.91 ± 4.79	63.44 ± 4.03
✓	✓		✓	65.69 ± 3.90	**62.83** ± 4.36	68.90 ± 10.24	65.22 ± 5.47	65.83 ± 3.91
✓	✓	✓		64.22 ± 3.87	59.97 ± 5.41	**76.80** ± 4.51	67.12 ± 3.31	64.84 ± 4.60
✓	✓	✓	✓	**66.27** ± 3.67	62.59 ± 4.40	73.07 ± 4.69	**67.25** ± 3.01	**66.62** ± 3.47

Important Modules. To inspect the effect of the important modules, we conduct experiments by disabling each of them without modifying other settings. The results are reported in Table 7.4. For ROI-wise attention Attn_{ROI}, subject-wise attention $\text{Attn}_{subject}$, and identity encoding (denoted as "ID enc" in the table), we disable them by simply removing these modules. When disabling "batch norm", we replace the batch normalization functions in the two-stream

attention by layer normalizations. The results demonstrate that Contrasformer with all important modules enabled achieves the best performance. Besides, the experiment of disabling $\text{Attn}_{subject}$ indicates that the outstanding recall of Contrasformer is mainly contributed by the subject-wise attention, demonstrating the effectiveness of extracting discriminative ROIs across subjects.

Table 7.5: Ablation study on the loss functions in Contrasformer on ABIDE dataset. The best result is highlighted in **bold**.

$\mathcal{L}_{cls}$	$\mathcal{L}_{entropy}$	$\mathcal{L}_{cluster}$	$\mathcal{L}_{contrast}$	Accuracy	Precision	Recall	micro-F1	ROC-AUC
✓		✓	✓	64.61 ± 3.93	61.79 ± 4.63	68.51 ± 9.36	64.45 ± 4.80	64.80 ± 3.69
✓	✓		✓	64.41 ± 4.34	60.36 ± 4.90	**75.57** ± 5.44	66.84 ± 2.86	64.99 ± 4.00
✓	✓	✓		59.51 ± 4.09	57.66 ± 6.84	58.25 ± 13.77	56.98 ± 8.30	59.50 ± 5.72
✓	✓	✓	✓	**66.27** ± 3.67	**62.59** ± 4.40	73.07 ± 4.69	**67.25** ± 3.01	**66.62** ± 3.47

Loss Functions. To verify the effectiveness of our proposed losses, we test our design of the loss functions by disabling them one by one. As shown in Table 7.5, the results demonstrate that all of those three auxiliary losses are effective in boosting the model performance. Besides, we find that the most important one is the contrastive loss. This observation indicates the necessity of introducing the constraint of node awareness.

Table 7.6: Comparison of time efficiency on ABIDE dataset. Epoch# reports the average converge epoch for 10-fold. Total time (h) was recorded for a single run (including training, validation, and test) with 10-fold CV. The last column shows the time cost relative to the most efficient method.

model	epoch#	total time (h)	percentage (%)
BrainNetCNN	263	0.91	110
LiNet	485	1.29	155
PRGNN	218	0.83	100
BrainGNN	230	4.58	552
BNTF	200	1.34	161
ContrastPool	293	6.33	763
Contrasformer (ours)	248	2.70	325

7.3.4 Time Efficiency

This chapter further compares the time efficiency of our proposed method with other models for brain networks in this subsection. Table 7.6 reports the number of epochs and the total

time needed (including training, validation and test) for these models. We can find from this experiment that both the number of epochs and the total runtime of Contrasformer are around medium among all baselines. Specifically, Contrasformer runs slower than simpler models of BrainNetCNN and PRGNN, but is much more efficient than state-of-the-art models BrainGNN and ContrastPool.

7.4 Summary

To overcome the hurdles of SPS noise and node-identity awareness, we introduce Contrasformer, a contrastive brain network Transformer. Through a contrast graph encoder with two-stream attention and a cross decoder with identity embedding, Contrasformer adaptively handles SPS noise, enhances node identity awareness, and captures group-specific patterns. Our model outperforms state-of-the-art methods in identifying neurological disorders across diverse datasets. The improvement over all the best models specifically designed for brain networks is up to 10.8%. Beyond superior performance, Contrasformer provides interpretable insights aligned with neuroscience literature. This chapter marks a significant advancement in harnessing Transformer models for fMRI-based brain network analysis, opening avenues for deeper understanding and diagnosis of neurological conditions.

Part III

Multi-atlas Brain Network Analysis

Chapter 8

Atlas-Integrated Distillation and Fusion Network

In this chapter[1], we propose an Atlas-Integrated Distillation and Fusion network (AIDFusion) to improve brain network classification using fMRI data. AIDFusion addresses the challenge of utilizing multiple atlases by employing a disentangle Transformer to filter out inconsistent atlas-specific information and distill distinguishable connections across atlases. It also incorporates subject- and population-level consistency constraints to enhance cross-atlas consistency. Additionally, AIDFusion employs an inter-atlas message-passing mechanism to fuse complementary information across brain regions. Experimental results on four datasets of different diseases demonstrate the effectiveness and efficiency of AIDFusion compared to state-of-the-art methods. A case study illustrates AIDFusion extract patterns that are both interpretable and consistent with established neuroscience findings.

8.1 Motivation

In the field of neuroscience, a key objective is to identify distinctive patterns associated with neurological disorders (e.g., Alzheimer's, Parkinson's, and Autism) by the brain networks [5]. Resting-state functional magnetic resonance imaging (fMRI) is widely employed among various neuroimaging techniques to characterize the connectivities among brain regions [200]. This results in brain networks where each node represents a specific brain region, referred to

[1]The work in this chapter has been submitted as "Multi-Atlas Brain Network Classification through Consistency Distillation and Complementary Information Fusion" for peer-review at Thirty-eighth Conference on Neural Information Processing Systems, 2024.

as a region of interest (ROI). Each edge indicates a pairwise correlation between the blood-oxygen-level-dependent (BOLD) signals of two ROIs [6], revealing the connectivity between brain regions and indicating which areas tend to be activated synchronously or exhibit correlated activities.

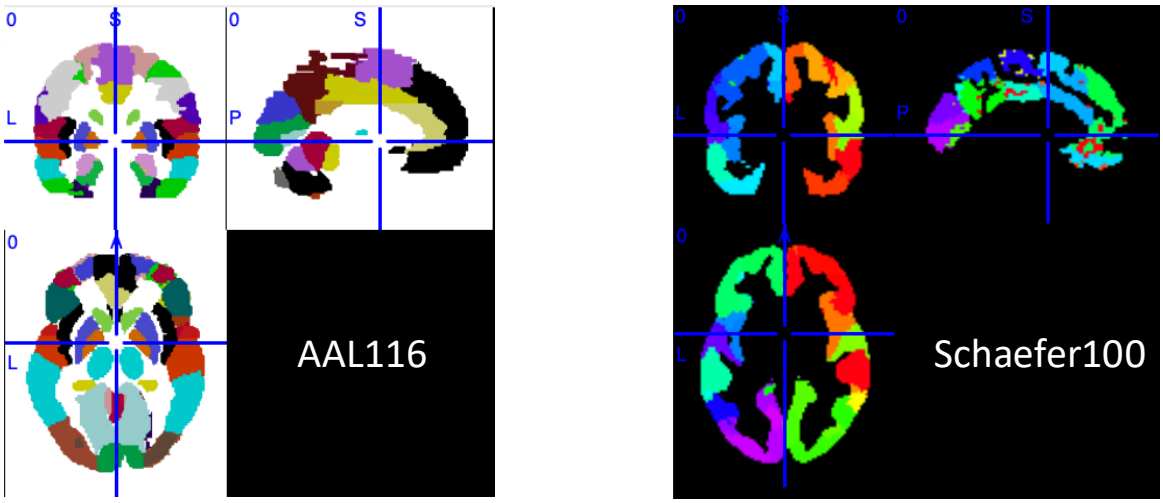


Fig. 8.1: AAL116 and Schaefer100 atlases. Each atlas is based on a different parcellation hypothesis.

Brain networks model neurological systems as graphs, allowing the use of graph-based techniques to understand their roles and interactions [2, 20, 201]. Constructing these brain networks involves using a specific atlas to parcellate the brain into ROIs. Various atlases based on different hypotheses of brain parcellation, such as anatomical and functional divisions, have been proposed to group similar fMRI regions and create ROIs [27–29]. Although proper brain parcellation is essential for detecting abnormalities in neurodegenerative disorders [30], there is no golden standard atlas for brain network classification. Relying on a single atlas for brain network analysis has two main drawbacks. First, some voxels may not be assigned to any specific ROI, potentially leading to the loss of important information. Second, as shown in Fig. 8.1, each atlas is based on a different parcellation hypothesis. The BOLD signal of an ROI is averaged from all voxels within it, possibly missing detailed information.

To address these limitations, recent works have proposed using multiple atlases with different parcellation modes to enhance multi-atlas brain network analysis. The framework of

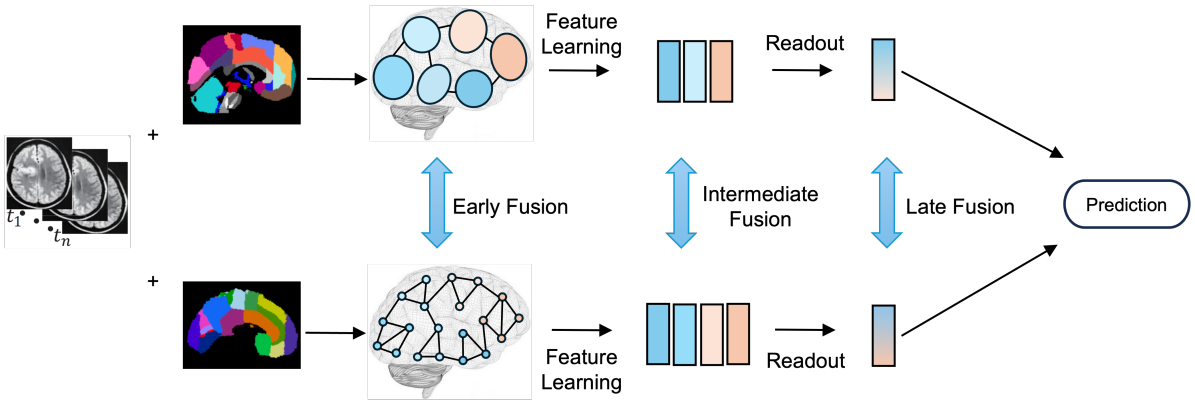


Fig. 8.2: The framework of multi-atlas brain network analysis.

multi-atlas brain network analysis is shown in Fig. 8.2. Some methods [31, 32] independently encode brain networks from various atlases and then aggregate the graph representations as a late feature fusion scheme for the final prediction. Another approach [33] incorporates early feature fusion by incorporating multi-atlas information from the raw data and using the fused feature for representation learning. However, these methods (1) neglect the need of consistency across atlases, potentially leading to the under-utilization of cross-atlas information; and (2) lack ROI-level information exchange throughout the entire representation learning process, which could hinder the models' ability to discern complementary information across different atlases.

In this chapter, we propose an Atlas-Integrated Distillation and Fusion network (AIDFusion) to address the aforementioned limitations by utilizing atlas-consistent information distillation and cross-atlas complementary information fusion. Specifically, AIDFusion introduces a disentangle Transformer to filter out inconsistent atlas-specific information and distill distinguishable connections across different atlases. Subject- and population-level consistency constraints are applied to enhance cross-atlas consistency. Furthermore, to facilitate the fusion of complementary information across ROIs in multi-atlas brain networks, AIDFusion employs an inter-atlas message-passing mechanism that leverages spatial information. Note that in our work, multiple atlases are applied to preprocessed images for parcellation, meaning our method is based on a single template. The difference between multi-atlas methods and multi-template methods are discussed in Appendix E.1.

8.2 Methodology

In this section, we provide a detailed exposition of the design of our proposed Atlas-Integrated Distillation and Fusion network (AIDFusion), depicted in Fig. 8.3. Two brain networks constructed with different atlases are separately processed in our model. In the following, we first formally defined the task of multi-atlas brain network classification. Afterward, we introduce the disentangle Transformer with identity embedding to remove inconsistent atlas-specific information (Section 8.2.2). We then describe the inter-atlas message-passing for spatial-based intense fusion of cross-atlas (Section 8.2.3). Finally, we discuss our design of the losses that enforce atlas-consistent information distillation with domain considerations (Section 8.2.4).

8.2.1 Problem Definition

Multi-atlas brain network classification aims to predict the distinct class of each subject by using various atlases for the same fMRI data. Given a dataset of labeled brain networks $\mathcal{D} = \{(\boldsymbol{X}^a, \boldsymbol{X}^b, y_X)\}$, where y_X is the class label of brain networks $\boldsymbol{X}^a$ and $\boldsymbol{X}^b$, the problem of brain network classification is to learn a predictive function f: $(\boldsymbol{X}^a, \boldsymbol{X}^b) \rightarrow y_X$, which maps input brain networks to the groups they belong to, expecting that f also works well on unseen brain networks.

8.2.2 Disentangle Transformer with Identity Embedding

Identity Embedding. In graph Transformer models, positional embedding is commonly used to encode the topological information of the graph. However, designs like distance-based, centrality-based, and eigenvector-based positional embeddings [75, 193, 194] are impractical for brain networks due to their high density (always fully connected). Correlation-based brain networks already contain sufficient positional information for ROIs, making general positional embeddings both costly and redundant. Instead, we propose a learnable identity embedding that adaptively learns a unique identity for each ROI, aligning nodes in the same ROI across the same atlas. This embedding assigns the same identity to nodes within the same ROI. As shown in Eq. (8.1), we introduce a parameter matrix $\boldsymbol{W}_{ID}$ to encode node identities alongside original node features $\boldsymbol{X}$, with $\delta(\cdot)$ denoting a multilayer perceptron (MLP).

$$\boldsymbol{H}_{ID} = \boldsymbol{X} + \delta(\boldsymbol{X} + \boldsymbol{W}_{ID}). \tag{8.1}$$

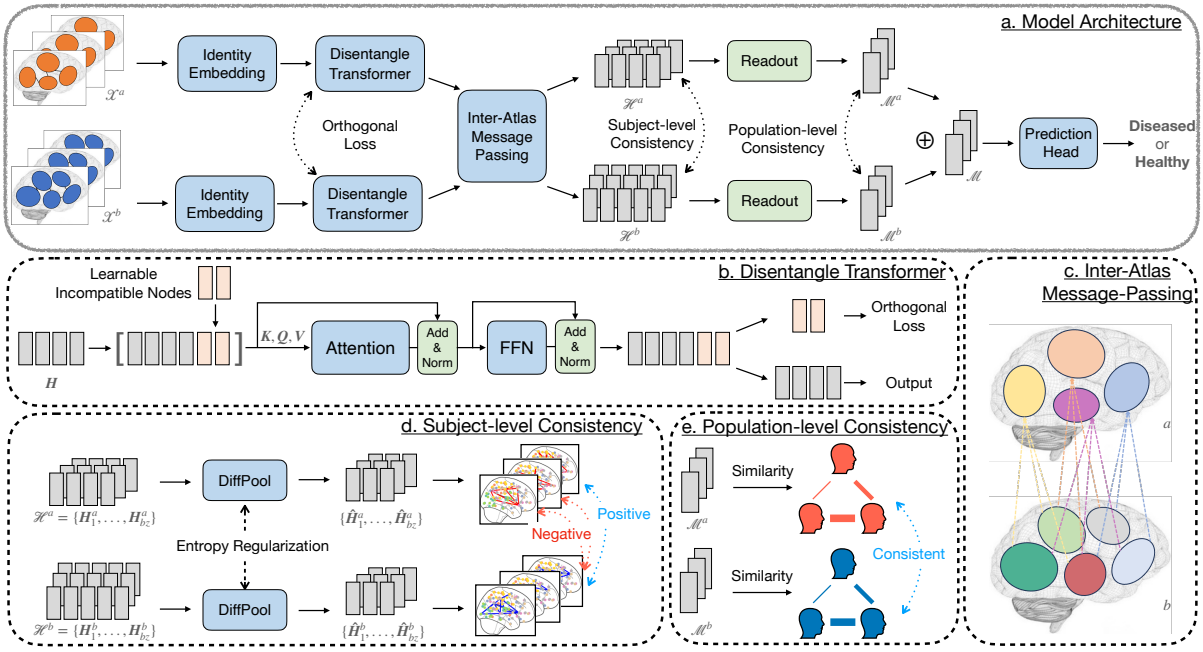


Fig. 8.3: The framework of AIDFusion for multi-atlas brain network classification.

Disentangle Transformer. Introducing learnable tokens in the input sequence of a Transformer has been a method used to capture global information. In natural language processing, Burtsev et al. [202] first utilized a learnable $[CLS]$ token to improve machine translation tasks. In computer vision, Darcet et al. [203] introduced register tokens to avoid recycling tokens from low-informative areas. Motivated by these prior works, we propose a disentangle Transformer to filter out inconsistent atlas-specific information by introducing incompatible nodes. We elaborate this module in Fig. 8.3b. Specifically, given an identity-encoded graph feature matrix $\boldsymbol{H}_{ID} \in \mathbb{R}^{n\times d}$, where n is the number of nodes and d is the hidden dimension, we add r learnable incompatible nodes $\boldsymbol{W}_{INC} \in \mathbb{R}^{r\times d}$ to the feature matrix:

$$\boldsymbol{H}' = \begin{bmatrix} \boldsymbol{H}_{ID} \\ \boldsymbol{W}_{INC} \end{bmatrix}, \tag{8.2}$$

where $\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$ denotes the append operation. To enforce each incompatible node captures different information, we initialize them using the Gram-Schmidt process [204] to ensure they are orthogonal to each other. Then the self-attention function [72] is applied to $\boldsymbol{H}' \in \mathbb{R}^{(n+r)\times d}$:

$$\text{Attn}(\boldsymbol{H}') = \text{norm}\left(\boldsymbol{H}' + \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathsf{T}}}{\sqrt{n+r}}\right)\boldsymbol{V}\right), \tag{8.3}$$

$$\boldsymbol{Q} = \boldsymbol{H}'\boldsymbol{W}_Q, \boldsymbol{K} = \boldsymbol{H}'\boldsymbol{W}_K, \boldsymbol{V} = \boldsymbol{H}'\boldsymbol{W}_V, \tag{8.4}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d\times d}$ are parameter matrices and norm$(\cdot)$ is a layer normalization.

In addition to the attention layer, a position-wise feed-forward network (FFN) with a layer normalization function is applied to each position to get the output node representations. The brain network of each atlas goes through a separate Disentangle Transformer. At the output of the disentangle Transformer, the incompatible nodes are discarded and only the ROI nodes are used.

Orthogonal Loss. As the brain networks derived from different atlases are based on the same fMRI data, we aim to ensure they contain similar information by filtering out inconsistent atlas-specific information. Therefore, we propose an orthogonal loss to enforce the representations of incompatible nodes to be orthogonal to each other across all atlases by minimizing their dot product:

$$\mathcal{L}_{orth} = \frac{1}{r}\sum \frac{\|\boldsymbol{W}^a_{INC} \cdot \boldsymbol{W}^b_{INC}\|}{\|\boldsymbol{W}^a_{INC}\| \cdot \|\boldsymbol{W}^b_{INC}\|}. \tag{8.5}$$

8.2.3 Inter-Atlas Message-Passing

The features at different atlases originate from totally different parcellation modes. Pulling those highly correlated features of two different atlases into a shared space allows their effective fusion. Existing literature on multi-atlas brain networks independently learns the representations of ROIs in each atlas without exchanging information across atlases [31, 33]. Additionally, the spatial relationship between ROIs in different atlases is neglected in these works. Our proposed AIDFusion enables inter-atlas message-passing between neighboring regions in different atlases by considering spatial information. Specifically, we use the spatial distance between the centroids of ROIs in different atlases to construct inter-atlas connections. As shown in Fig. 8.3c, we utilize the k-nearest-neighbor (kNN) algorithm to connect each ROI to k ROIs from the other atlas.

Specifically, given two atlases a and b with n^a and n^b ROIs, we denote the 3D coordinate of the i-th and j-th ROI of them as $\boldsymbol{C}^a_i$ and $\boldsymbol{C}^b_j$, respectively. The distance matrix $\boldsymbol{Dis}^{ab} \in \mathbb{R}^{n_a\times n_b}$ is computed by Euclidean distance $\boldsymbol{Dis}^{ab}_{ij} = \text{distance}(\boldsymbol{C}^a_i, \boldsymbol{C}^b_j)$. A mask matrix $\boldsymbol{Mask}$ is then generated, where $\boldsymbol{Mask}^{ab}_{ij} = 1$ if $j \in topk(\boldsymbol{Dis}^{ab}_i)$, and 0 otherwise. Afterwards, the inter-atlas adjacency matrix is defined as:

$$\boldsymbol{A}^{ab} = \begin{bmatrix} 0 & \boldsymbol{Mask}^{ab} \\ \boldsymbol{Mask}^{ba} & 0 \end{bmatrix}. \tag{8.6}$$

We summarize this process in the following algorithm.

Algorithm 2 Construction of the inter-atlas adjacency matrix $\boldsymbol{A}^{ab}$.

Input: The 3D ROI coordinates $\boldsymbol{C}^a$ and $\boldsymbol{C}^b$ of atlases a and b;
Output: $\boldsymbol{A}^{ab}$;

$$\boldsymbol{D}^{ab}_{ij} = \text{distance}(\boldsymbol{C}^a_i, \boldsymbol{C}^b_j);$$
$$\boldsymbol{M}^{ab}_{ij} = \begin{cases} 1 & \text{if } j \in \text{topk}(\boldsymbol{D}^{ab}_i) \\ 0 & \text{otherwise} \end{cases};$$
$$\boldsymbol{D}^{ba}_{ij} = \text{distance}(\boldsymbol{C}^b_i, \boldsymbol{C}^a_j);$$
$$\boldsymbol{M}^{ba}_{ij} = \begin{cases} 1 & \text{if } j \in \text{topk}(\boldsymbol{D}^{ba}_i) \\ 0 & \text{otherwise} \end{cases};$$
$$\boldsymbol{A}^{ab} = \begin{bmatrix} 0 & \boldsymbol{M}^{ab} \\ \boldsymbol{M}^{ba} & 0 \end{bmatrix}.$$

Note that we only construct inter-atlas connections without considering intra-atlas connections since the information exchange within the same atlas has already proceeded in previous disentangle Transformer. Afterwards, an adjacency matrix $\boldsymbol{A}^{ab} \in \{0,1\}^{(n^a+n^b)\times(n^a+n^b)}$ is obtained and used for graph convolution [152]:

$$\text{GCN}(\boldsymbol{A}^{ab}, \boldsymbol{H}^{ab}) = \sigma\left(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}^{ab}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}^{ab}\boldsymbol{W}_{GC}\right), \tag{8.7}$$

where σ is the activation function (e.g., ReLU), $\boldsymbol{D}$ is the degree matrix of $\boldsymbol{A}^{ab}$, $\boldsymbol{H}^{ab} \in \mathbb{R}^{(n^a+n^b)\times d}$ is the combined node representation matrix for the two atlases, and $\boldsymbol{W}_{GC}$ is the learnable weight matrix of the GCN layer. An example of the adjacency matrix $\boldsymbol{A}^{ab}$ that used for inter-atlas message-passing is shown and discussed in Appendix E.2.

8.2.4 Subject- and Population-level Consistency

Subject-level Consistency. To ensure the high-level consistency for the two brain networks from different atlases, we introduce a contrastive loss on the subject level. First, we apply DiffPool [10] to each atlas to capture higher-level patterns. The DiffPool contains two GCN layers.

GCN_{pool} is used to learn a cluster assignment matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n'}$ as shown in Eq. (8.8). Herein, n' is a pre-defined number of clusters controlled by a hyperparameter named the pooling ratio. The other GCN_{emb} is used to obtain the embedded node feature matrix $\boldsymbol{Z} \in \mathbb{R}^{n \times d}$ as shown in Eq. (8.9). Both these two GCNs are defined similarly with Eq. (8.7). We sparsify the connectivity matrices $\boldsymbol{X}$ by keeping top 20% correlations and use them as the adjacency matrices $\boldsymbol{A}$ in these two GCNs, to avoid over-smoothing. The feature matrices of two atlas $\boldsymbol{H}$ are obtained from the output of inter-atlas message-passing.

$$\boldsymbol{S} = \text{softmax}\left(\text{GCN}_{pool}(\boldsymbol{A}, \boldsymbol{H})\right). \tag{8.8}$$

$$\boldsymbol{Z} = \text{GCN}_{emb}(\boldsymbol{A}, \boldsymbol{H}). \tag{8.9}$$

Once we obtain the cluster assignment matrix $\boldsymbol{S}$ and the embedded node feature matrix $\boldsymbol{Z}$, we generate a new feature matrix $\hat{\boldsymbol{H}} \in \mathbb{R}^{n' \times d}$ by $\hat{\boldsymbol{H}} = \boldsymbol{S}^{\mathsf{T}}\boldsymbol{Z}$. This coarsening process can reduce the number of nodes to get higher-level node representations. To avoid GNN treating each ROI and each node cluster equally, we adopt an entropy regularization to the assignment matrices of each atlas:

$$\mathcal{L}_E = \frac{1}{n'}\sum_{i=1}^{n'}\left(\text{entropy}(\boldsymbol{S}^a[i,:]) + \text{entropy}(\boldsymbol{S}^b[i,:])\right), \text{entropy}(\boldsymbol{p}) = -\sum_{j=1}^{n'} \boldsymbol{p}_j \log(\boldsymbol{p}_j). \tag{8.10}$$

We elaborate on the module of subject-level consistency in Fig. 8.3d. Through two DiffPool layers, we produce high-quality representations for each atlas by extracting high-level node representations. Then we are able to apply a contrastive loss to them by considering representations from the same subject as positive pairs $\mathcal{P}^{pos} = \{(\hat{\boldsymbol{H}}_i^a, \hat{\boldsymbol{H}}_i^b)\} : i = 1, \ldots, bz\}$ and representations from different subjects as negative pairs $\mathcal{P}^{neg} = \{(\hat{\boldsymbol{H}}_i^a, \hat{\boldsymbol{H}}_{\neg i}^b) : i = 1, \ldots, bz\}$:

$$\mathcal{L}_{SC} = -\log \frac{\sum \sum_{(\hat{\boldsymbol{H}}_i^a, \hat{\boldsymbol{H}}_i^b) \in \mathcal{P}^{pos}} \exp(\text{sim}(\hat{\boldsymbol{H}}_i^a, \hat{\boldsymbol{H}}_i^b)/\tau)}{\sum \sum_{(\hat{\boldsymbol{H}}_i^a, \hat{\boldsymbol{H}}_{\neg i}^b) \in \mathcal{P}^{neg}} \exp(\text{sim}(\hat{\boldsymbol{H}}_i^a, \hat{\boldsymbol{H}}_{\neg i}^b)/\tau)}, \tag{8.11}$$

where τ is a temperature hyper-parameter to control the smoothness of the probability distribution [55], bz is the batch size, and $\text{sim}(\cdot)$ denotes the cosine similarity function that is applied to the same row in the two matrices.

Population-level Consistency. The readout function $\boldsymbol{m} = \text{READOUT}(\boldsymbol{H})$ is an essential component of learning the graph-level representations $\boldsymbol{m} \in \mathbb{R}^d$ for brain network analysis (e.g.,

classification), which maps a set of learned node-level embeddings to a graph-level embedding. To further constrain the consistency for graph representations across different atlases, we introduce a mean squared error (MSE) loss on the population level. As shown in Fig. 8.3e, a population graph $\boldsymbol{G}$ is constructed by computing the similarity of each two subjects' graph representations in the same atlas. The intuition here is we aim to maintain the relationship of subjects across atlases, instead of directly enforcing graph representations of two atlases to be the same. Such loss is formulated as follows:

$$\mathcal{L}_{PC} = \frac{1}{bz}\sum(\boldsymbol{G}^a - \boldsymbol{G}^b)^2, \boldsymbol{G}[i,j] = \text{sim}(\boldsymbol{m}_i, \boldsymbol{m}_j), \boldsymbol{m}_i, \boldsymbol{m}_j \in \mathcal{M}, \tag{8.12}$$

where $\mathcal{M}$ is the set of graph representations in a batch.

Total Loss. The model is supervised by a commonly-used cross-entropy loss $\mathcal{L}_{cls}$ [167] for graph classification. The total loss is computed by:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \lambda_1 * \mathcal{L}_{SC} + \lambda_2 * \mathcal{L}_{PC} + \lambda_3 * \mathcal{L}_{E} + \lambda_4 * \mathcal{L}_{orth}, \tag{8.13}$$

where λ_1, λ_2, λ_3 and λ_4 are trade-off hyperparameters for balancing different losses.

8.3 Experimental Results

In this section, we first detail the baseline models of our experiments. We then assess the performance of AIDFusion in comparison with 13 baseline models. We discuss the influence of using more atlases with different hypotheses and resolutions afterward. We also present two case studies to provide the domain interpretation of the generated contrast graph and discuss the generalization ability of our method. We further conduct ablation studies to analyze the effects of the components in AIDFusion. In the end, we report the time efficiency of both our method and baseline models.

8.3.1 Baseline Models

We use 7 single-atlas methods and 6 multi-atlas methods as baselines to evaluate our proposed AIDFusion, including: (1) Conventional machine learning (ML) models: Logistic Regression (**LR**) and Support Vector Machine Classifier (**SVM**) from scikit-learn [180]. These models take the flattened upper-triangle connectivity matrix as vector input, instead of using the brain

network. (2) General-purposed GNNs: **GCN** [152], a mean pooling baseline with a graph convolution network as a message-passing layer and **Transformer** [72], a graph Transformer with mean pooling by taking the connectivity matrix as input. (3) Single-Atlas Models tailored for brain networks: **BrainNetCNN** [20], the pioneering CNN regressor for connectome data; **MG2G** [80], a two-stage method with an unsupervised stochastic graph embedding model; and **ContrastPool** [179], a node clustering pooling using a dual-attention block for domain-specific information capturing. (4) Multi-atlas models: **MultiLR**, multi-atlas version of LR, concatenate the flatten feature of multiple atlases as input; **MultiSVM**, multi-atlas version of SVM, similar with MultiLR; **MGRL** [31], a mean pooling baseline by individual GCN encoder with late fusion; **MGT**, a multi-atlas version of Transformer with the same fusion mechanism as MGRL; **METAFormer** [32], a multi-atlas enhanced Transformer with self-supervised pre-training; and **LeeNet** [33] a multi-atlas GCN approach with early-late fusion.

The implementation detail of our experiments is given in Appendix E.3.

8.3.2 Main Results

We report the classification accuracy on 4 brain network datasets over 10-fold cross-validation in Table 8.1. For certain diseases, the effectiveness/informativeness of different atlases is different. On Matai, all the 7 baselines attain better performance with AAL116 than with Schaefer100. On ADNI, 6 out of 7 baselines also perform better with AAL116. In contrast on ABIDE, 5 out of 7 baselines achieve better results with Schaefer100 than with AAL116. It demonstrates the importance of using multi-atlas for brain network analysis instead of relying on one specific atlas. Moreover, it is evident that the multi-atlas baselines with a simple late fusion mechanism (MGRL and MGT) outperform their respective single-atlas models (GCN and Transformer). This highlights the effectiveness of multi-atlas approaches in enhancing the performance of base models. However, conventional ML models (MultiLR and MultiSVM) fail to outperform their single-atlas versions in some cases, possibly due to their inability to effectively utilize multi-atlas features with simple concatenate fusion.

We can also observe that our proposed AIDFusion consistently outperforms not only all single-atlas methods but also state-of-the-art multi-atlas methods across all datasets. Specifically, AIDFusion achieves improvements over all multi-atlas methods on these four datasets by

Table 8.1: Graph Classification Results (Average Accuracy ± Standard Deviation) over 10-fold-CV. The first and second best results on each dataset are highlighted in **bold** and underline.

atlas	model	ABIDE	ADNI	PPMI	Mātai
Schaefer100	LR	64.81 ± 3.70	61.97 ± 4.24	56.48 ± 6.76	60.00 ± 20.00
	SVM	64.41 ± 5.09	61.52 ± 4.95	63.21 ± 8.62	56.67 ± 17.00
	GCN	60.19 ± 2.96	60.40 ± 4.89	54.02 ± 9.06	56.67 ± 17.00
	Transformer	59.90 ± 3.77	63.64 ± 2.61	59.33 ± 5.68	60.00 ± 20.00
	BrainNetCNN	65.75 ± 3.24	60.48 ± 3.29	57.33 ± 10.32	61.67 ± 13.33
	MG2G	64.41 ± 2.16	63.64 ± 5.10	55.45 ± 10.24	61.67 ± 19.79
	ContrastPool	65.01 ± 3.84	65.67 ± 6.64	64.00 ± 6.63	61.67 ± 13.02
AAL116	LR	63.80 ± 3.00	64.06 ± 1.80	56.00 ± 7.79	66.67 ± 21.08
	SVM	65.72 ± 3.30	63.40 ± 1.90	64.12 ± 5.69	65.00 ± 20.34
	GCN	60.10 ± 5.74	61.24 ± 2.47	53.14 ± 8.82	65.00 ± 21.67
	Transformer	60.88 ± 4.39	63.27 ± 2.79	61.24 ± 7.22	63.33 ± 24.49
	BrainNetCNN	64.58 ± 6.29	62.52 ± 2.91	51.19 ± 9.24	66.67 ± 18.33
	MG2G	62.99 ± 4.01	64.41 ± 2.52	59.71 ± 9.11	70.00 ± 19.44
	ContrastPool	64.70 ± 3.26	66.33 ± 4.10	63.56 ± 7.90	65.00 ± 20.82
Schaefer100 + AAL116	MultiLR	65.23 ± 5.13	64.99 ± 2.40	55.00 ± 6.25	56.67 ± 24.94
	MultiSVM	64.31 ± 5.24	65.21 ± 2.74	63.60 ± 7.66	58.33 ± 17.08
	MGRL	61.56 ± 4.90	62.74 ± 3.55	54.55 ± 10.67	68.33 ± 18.93
	MGT	63.32 ± 3.90	63.99 ± 4.34	62.14 ± 9.90	65.00 ± 22.91
	METAFormer	61.27 ± 4.05	66.52 ± 2.63	54.02 ± 8.81	61.67 ± 25.87
	LeeNet	61.28 ± 3.12	64.63 ± 1.34	60.74 ± 4.39	58.33 ± 17.08
	AIDFusion (ours)	**66.35** ± 3.26	**67.57** ± 2.04	**66.00** ± 4.71	**75.00** ± 13.44

up to 9.76% ((75.00% - 68.33%) / 68.33% = 9.76% on Mātai). Our model gains larger performance improvement on small datasets (PPMI and Mātai) than on large datasets (ABIDE and ADNI), which meets the intuition that information utilization tends to be more critical in applications with smaller sample sizes. Moreover, the results demonstrate that AIDFusion tends to have lower standard deviations compared to other multi-atlas models, indicating the robustness of AIDFusion. This robustness is particularly desirable in medical applications where consistency and reliability are crucial.

In addition to accuracy, we also report other evaluation metrics, including precision, recall, micro-F1, and ROC-AUC, for all the multi-atlas deep models on the ABIDE dataset. As displayed in Table 8.2, AIDFusion performs the best across all these metrics except for precision. We observe that compared to other baselines, our AIDFusion can significantly improve recall without compromising precision. Moreover, in medical diagnostics, it is crucial to ensure that all individuals with a certain condition are correctly identified, even if it leads to some false

Table 8.2: Results of more evaluation metrics on ABIDE dataset. The best result is highlighted in **bold**.

	Precision	Recall	micro-F1	ROC-AUC
MGRL	59.90 ± 6.42	60.03 ± 6.31	59.71 ± 5.10	61.39 ± 5.08
MGT	60.33 ± 4.78	69.29 ± 6.74	64.21 ± 3.62	63.60 ± 3.80
METAFormer	59.33 ± 4.05	61.91 ± 7.79	60.20 ± 4.26	61.31 ± 3.94
LeeNet	**64.30** ± 5.24	43.04 ± 6.09	51.23 ± 4.74	60.44 ± 3.06
AIDFusion (ours)	62.25 ± 3.00	**74.80** ± 4.38	**67.90** ± 3.12	**66.73** ± 3.25

positives. Missing a true positive (failing to diagnose a disease) can have severe consequences, while false positives can be further examined or retested. Therefore, models with higher recall rates, such as our AIDFusion, are more suitable for real-life medical auxiliary diagnosis. Further hyperparameter sensitivity analysis is provided in Appendix E.4.

8.3.3 Results with More Atlases

To further evaluate the effectiveness of AIDFusion with other atlases, we conducted experiments using an additional atlas, HO48 [28], on the ADNI dataset. The results, presented in Table 8.3 indicate that the proposed AIDFusion achieves the best performance across all four atlas settings. Notably, increasing the number of atlases does not necessarily enhance model performance. In some cases, using all three atlases yields lower accuracy compared to the combination of Schaefer100 and AAL116. Additionally, the choice of atlas combination is crucial for multi-atlas methods. In dual-atlas experiments, combining two atlases with a similar number of ROIs (Schaefer100 and AAL116) can mutually enhance and significantly improve model performance.

To further explore how the resolution of atlases will influence the performance of our model, we conduct experiments for atlases with various numbers of ROIs. The Schaefer atlas allows adjusting the resolution of ROIs (e.g., from 100 to 1000). We selected Schaefer100 for detailed study because a previous study [6] found that using 100 ROIs with the Schaefer atlas usually performs better than using more ROIs. To verify this conclusion in multi-atlas brain network classification, we conducted experiments using AAL116 combined with Schaefer200, Schaefer500, and Schaefer1000. Results showed that AAL116 combined with Schaefer100 achieves the best results. It is also interesting for us to explore using both atlases of around

Table 8.3: Results of more atlases results on ADNI dataset. The best results for each atlas setting are highlighted in **bold**.

Atlas			model	acc ± std
Schaefer100	AAL116	HO48		
			MGRL	62.74 ± 3.55
			MGT	63.99 ± 4.34
✓	✓		METAFormer	66.52 ± 2.63
			LeeNet	64.63 ± 1.34
			AIDFusion	**67.57** ± 2.04
			MGRL	64.48 ± 1.68
			MGT	60.48 ± 1.91
	✓	✓	METAFormer	64.48 ± 1.68
			LeeNet	64.48 ± 1.58
			AIDFusion	**65.99** ± 2.62
			MGRL	57.92 ± 2.82
			MGT	60.64 ± 3.52
✓		✓	METAFormer	65.23 ± 3.25
			LeeNet	63.42 ± 1.82
			AIDFusion	**65.91** ± 1.80
			MGRL	56.63 ± 4.66
			MGT	62.89 ± 1.28
✓	✓	✓	METAFormer	66.33 ± 2.80
			LeeNet	64.40 ± 1.71
			AIDFusion	**66.59** ± 1.77

200 nodes (or around 1000 nodes). We will leave such exploration about multi-scale brain networks in the future.

8.3.4 Model Interpretation

In neurodegenerative disorder diagnosing, identifying salient ROIs/connections associated with predictions as potential biomarkers is crucial. In this study, we utilize attention scores from the Transformer layer to generate heat maps for brain networks to interpret our model. We visualize these attention maps using the Nilearn toolbox [145]. Fig. 8.4 presents attention maps for two atlases, where higher attention values mean better classification potential for AD (from the ADNI dataset). We utilized 7 networks [205] to assess the connections between our highlighted ROIs and major networks potentially involved with disorders. ROIs from the AAL that do not overlap with these seven networks are excluded from the heat maps. The top 10 ROIs

Table 8.4: Results of more atlases with different resolutions on ABIDE dataset. The best results for each atlas setting are highlighted in **bold**.

Atlas 1	Atlas 2	acc ± std
AAL116	Schaefer100	**66.35** ± 3.26
AAL116	Schaefer200	65.03 ± 5.10
AAL116	Schaefer500	64.72 ± 4.80
AAL116	Schaefer1000	63.15 ± 2.80

with the highest attention values are displayed in the brain view. As depicted in the attention maps, attention maps of both Schaefer and AAL atlases identify common connections between the visual network (VIS) and the dorsal attention network (DAN), recognized as key connectivities in AD research [185, 206]. Additionally, atlas-specific connections are highlighted. For example, the attention map of Schaefer atlas emphasizes connections within the default mode network (DMN) corresponding with the observations of Damoiseaux et al. [207]. Findings on the attention map on AAL are consistent with Agosta et al. [206], showing that AD is associated with connectivities in VIS, especially in frontal networks. These findings suggest that AIDFusion effectively captures complementary information from different atlases. We also find some highlighted ROIs that diverge from conventional neuroscientific understanding. For example, connections between VIS and somatomotor network (SMN) have a high attention weight in AIDFusion on Schaefer atlas, which may imply AD is related to the function of defining the targets of actions and providing feedback for visual activation. This insight has not been identified by existing literature.

In the ASD analysis using the ABIDE dataset (Fig. 8.5), AIDFusion identifies common connections within the lingual gyrus of the VIS network in both Schaefer100 and AAL116. This aligns with existing ASD studies which suggest a greater reliance on visual perceptual processing and more effortful top-down control during semantic processing in ASD [208]. Besides, for Schaefer, AIDFusion emphasizes ROIs in the DAN, particularly the connection between the right posterior and VIS, consistent with Koshino et al. [209] who found lower activation in ASD subjects in the inferior left prefrontal area (verbal processing and working memory) and the right posterior temporal area (theory of mind processing). In the AAL analysis, connections in the DMN and the frontoparietal control network (FPCN) are highlighted, supporting the understanding that (1) dysfunctions in DMN nodes and their interactions contribute to difficulties of ASD in attending to socially relevant stimuli [210], and (2) the ASD

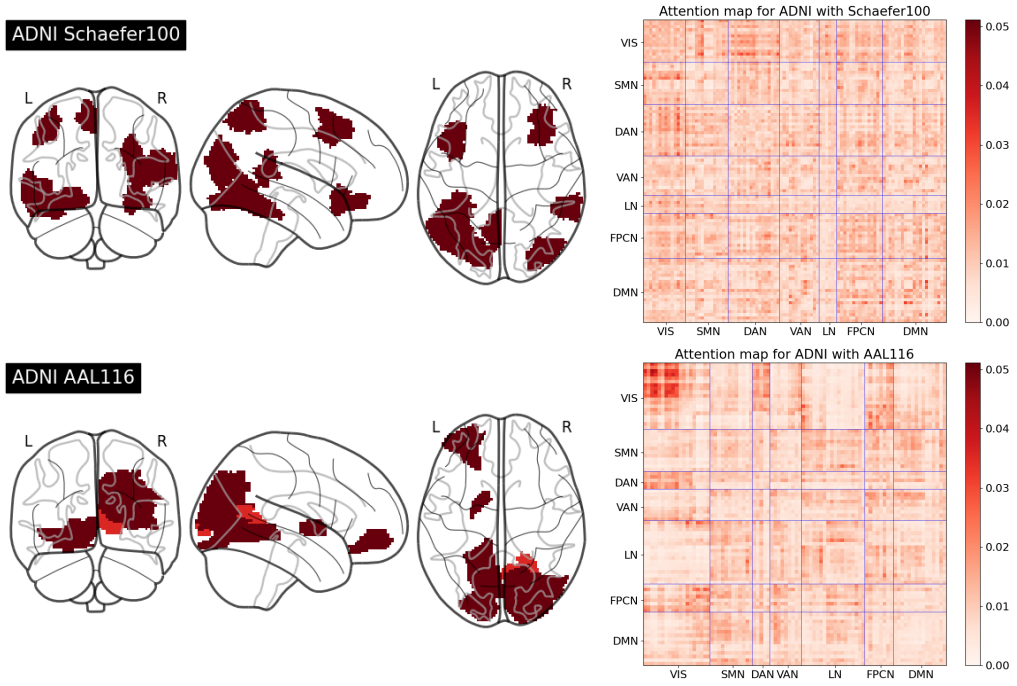


Fig. 8.4: Visualization for attention maps on ADNI. VIS = visual network; SMN = somatomotor network; DAN = dorsal attention network; VAN = ventral attention network; LN = limbic network; FPCN = frontoparietal control network; DMN = default mode network.

group shows reduced lateral frontal activity and diminished hippocampal connectivity, especially between the hippocampus and FPCN regions [211]. These findings clarify why the features identified by AIDFusion are distinctive for ASD biomarkers.

8.3.5 Ablation Study

Table 8.5: Ablation study on the key components of AIDFusion on ADNI, with the best result **bold**.

Backbone	IA-MP	Subject-level Consistency	Population-level Consistency	acc ± std
TF				63.99 ± 4.34
TF	✓	✓	✓	66.82 ± 1.25
Disen TF		✓	✓	66.58 ± 1.72
Disen TF	✓		✓	66.37 ± 1.56
Disen TF	✓	✓		65.91 ± 2.08
Disen TF	✓	✓	✓	**67.57** ± 2.04

To inspect the effect of the key components in AIDFusion, we conduct experiments by dis-

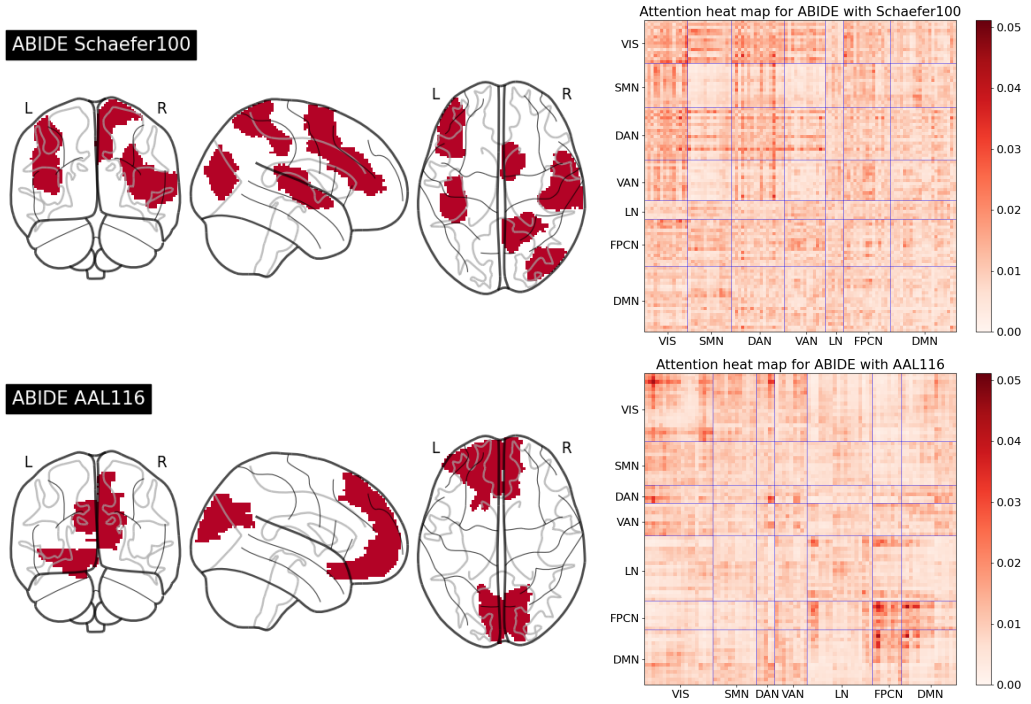


Fig. 8.5: Visualization for attention maps on ABIDE. VIS = visual network; SMN = somatomotor network; DAN = dorsal attention network; VAN = ventral attention network; LN = limbic network; FPCN = frontoparietal control network; DMN = default mode network.

abling each of them without modifying other settings. The results on ADNI dataset are reported in Table 8.5. For inter-atlas message-passing (denoted as "IA-MP" in the table), subject-level consistency and population-level consistency, we disable them by simply removing these modules. When disabling "Disen TF", we replace the disentangle Transformer and the identity embedding with a vanilla Transformer backbone (denoted as "TF" in the table). When disabling all key components (the first row in the table), our model will degenerate to MGT in Table 8.1. The results demonstrate that AIDFusion with all important modules enabled achieves the best performance. The component that affect the performance most is the population-level consistency. Besides, all variants of the proposed AIDFusion outperform the MGT baseline, demonstrating the effectiveness of our model design.

We further explore the function of incompatible nodes by visualizing the attention map of ADFusion w/o incompatible nodes. The attention maps are shown in Fig. 8.6. We can observe that, when not using incompatible nodes, the attentions of two atlases (in the right column) are remarkably imbalanced. Attentions on Schaefer are much higher than those on AAL. Besides, the attention map of AAL exhibits over-smoothing and no highlighted network is found, which

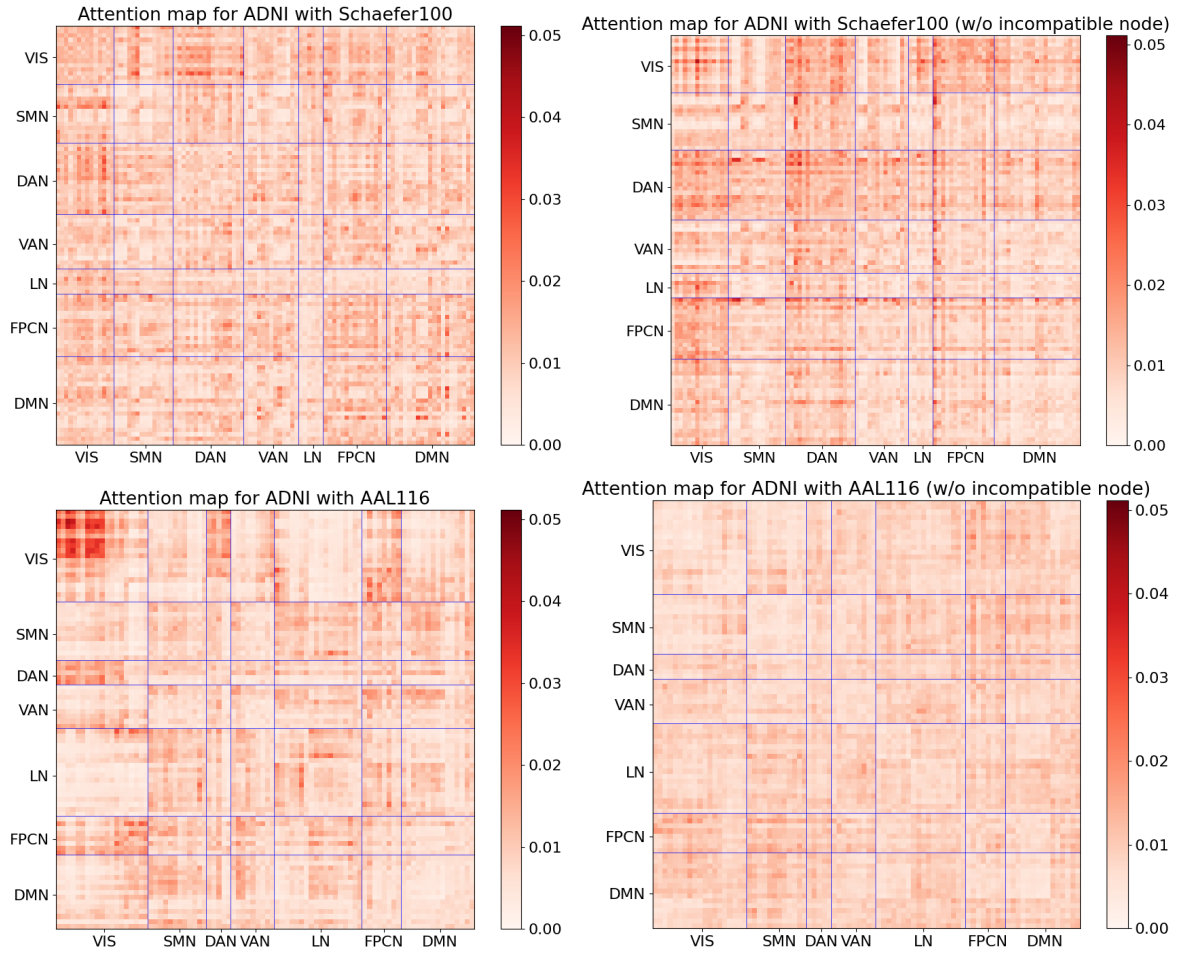


Fig. 8.6: Visualization for attention maps of AIDFusion w/ and w/o incompatible nodes.

indicates the model is not able to extract the distinguishable connections. This case study demonstrates that the incompatible nodes enable the model to filter out the inconsistent atlas-specific information.

8.3.6 Time Efficiency

We conducted an experiment to compare the total runtime cost of AIDFusion with other multi-atlas baselines. The results, reported in Table 8.6, demonstrate that AIDFusion requires dramatically fewer epochs to converge, resulting in significantly less time spent on ABIDE, ADNI, and PPMI datasets. For the Mātai dataset, AIDFusion's time cost is still comparable with the other baselines. Besides, since AIDFusion does not contain any repeat layers as other baselines

do, it has fewer parameters and thus results in higher efficiency. This showcases the efficiency of the proposed AIDFusion.

Table 8.6: Time efficiency analysis. Total time (h) was recorded with a single run (including training, validation, and test) with 10-fold CV.

	ABIDE		ADNI		PPMI		Mātai		#Param
	Time (h)	#Epoch	Time (h)	#Epoch	Time (h)	#Epoch	Time (h)	#Epoch	
MGRL	0.56	261.9 ± 0.7	0.91	262.7 ± 0.8	0.15	272.2 ± 8.0	0.09	291.2 ± 14.7	378k
MGT	0.78	263.9 ± 2.4	0.89	108.2 ± 1.1	0.08	134.3 ± 10.9	0.18	266.4 ± 4.2	273k
METAFormer	1.73	263.2 ± 1.5	1.47	268.3 ± 2.2	0.16	266.5 ± 4.7	0.12	270.4 ± 3.6	1886k
LeeNet	1.56	200.0 ± 0.0	1.76	200.0 ± 0.0	0.19	200.0 ± 0.0	**0.07**	200.0 ± 0.0	526k
AIDFusion (ours)	**0.12**	**48.5** ± 19.0	**0.26**	**64.6** ± 14.1	0.03	**34.9** ± 12.0	0.10	**119.5** ± 13.8	**235k**

8.4 Summary

In this chapter, we presented the Atlas-Integrated Distillation and Fusion network (AIDFusion), a novel approach to multi-atlas brain network classification. The disentangle Transformer mechanism, combined with inter-atlas message-passing and consistency constraints, effectively integrates complementary information across different atlases and ensures cross-atlas consistency at both the subject and population levels. Our extensive experiments on four fMRI datasets demonstrate that AIDFusion outperforms state-of-the-art methods in terms of classification accuracy and efficiency. Moreover, the patterns identified by AIDFusion align well with existing domain knowledge, showcasing the model's potential for providing interpretable insights into neurological disorders. For now, the discussion of AIDFusion is restricted to fMRI with 3 atlases, our future work will explore extending AIDFusion to other neuroimaging modalities and further studying the model's capability of selecting atlases for different diseases. We hope our work inspires further research in multi-atlas brain network analysis and demonstrates its significance in real-world applications, such as early diagnosis and personalized treatments for neurodegenerative diseases.

Chapter 9

Conclusions and Future Work

9.1 Conclusion

In conclusion, this thesis addresses the evolving field of network neuroscience with a specific focus on functional magnetic resonance imaging (fMRI) and its critical role in understanding brain functions and neurodegenerative conditions. We have tackled the challenges associated with constructing brain networks from fMRI data and have identified the limitations of applying general-purpose Graph Neural Networks (GNNs) to this unique data domain.

To bridge these gaps and facilitate further research, this thesis presents a comprehensive collection of resting-state functional brain networks, demonstrating their quality and utility as valuable benchmarks for the community. The outcomes in this thesis, including the class-aware representation refinement framework (CARE), the contrastive graph pooling method (ContrastPool), and the contrastive brain network Transformer (Contrasformer), offer innovative and effective solutions for graph-based analysis of fMRI data. Besides using one single atlas for brain network analysis, we also proposed a multi-atlas solution (AIDFusion) to capture the whole picture of fMRI data.

The experimental results not only highlight the effectiveness and efficiency of these approaches but also emphasize their interpretability, aligning with domain knowledge found in neuroscience literature. This thesis promotes interdisciplinary research at the intersection of network neuroscience, machine learning, and graph analytics, providing a foundation for advancing our understanding of neurodegenerative conditions and other clinical applications in the context of brain networks.

9.2 Future Work

In our future research, we aim to address several critical challenges in brain network analysis to enhance our understanding and diagnosis of neurological disorders. First, we will explore comprehensive multi-modality brain networks by developing effective unsupervised learning models that integrate structural and functional data, addressing the lack of public benchmark datasets and improving generalizability and explainability. Second, we will tackle the out-of-distribution (OOD) problem in brain network analysis, caused by site differences in data collection, by developing novel algorithms that consider feature information and brain network characteristics like node alignment. Third, we plan to advance multi-scale brain network analysis by creating public benchmarks and leveraging inter-scale information and class-based features to capture the brain's hierarchical organization more effectively. Finally, we will delve into dynamic brain network modeling, incorporating temporal dynamics from resting-state fMRI and other time-series data to provide a more comprehensive understanding of neurological processes and brain connectivity. Together, these efforts will significantly advance the field of computational network neuroscience, offering new insights and improving early diagnosis and intervention for neurological disorders.

9.2.1 Multi-modal Studies

Traditional methods either focus on a single type of brain data (structural or functional) or adopt a supervised learning paradigm, limiting the comprehensiveness and generalizability of the analysis. We plan to explore comprehensive multi-modality brain networks and develop effective and generalizable computational models using an unsupervised approach.

Current multi-modality brain network analysis [93–97] faces three key challenges: (1) the lack of publicly available benchmark data, (2) the oversight of brain network characteristics, and (3) limited generalizability and explainability. To address these challenges, we will create the first publicly available multi-modality brain network benchmark dataset encompassing four brain conditions. This dataset will be derived from structural (DTI) and functional (rs-fMRI) neuroimages. Subsequently, we will develop unsupervised learning methods to analyze these networks, incorporating unique brain network characteristics and fusing multi-modality information. Moreover, the interpretability of multi-modal brain networks remains a challenge [**?**].

Developing methods that not only improve classification or prediction performance but also provide insights into the underlying brain mechanisms is crucial. This could be achieved through the use of explainable AI techniques that highlight the brain regions or network features most relevant to the task at hand. These methods aim to extract intrinsic and inherent structures in brain networks, leading to a deeper understanding of brain connectivity in neurological disorders.

Our research constitutes a timely and novel contribution to the critical intersection of machine learning and neuroscience. It holds significant intellectual merit by advancing unsupervised learning in multi-modality brain network analysis, uncovering novel insights into brain organization and function that traditional methods may have overlooked. Additionally, it aims to revolutionize state-of-the-art analytic pipelines in computational network neuroscience, leading to new knowledge and scientific findings. The outcomes of this research will benefit individual and societal health by enabling the early diagnosis of neurological disorders, making early intervention possible.

9.2.2 Out-of-distribution for Brain Networks

Machine learning has advanced significantly in recent years. However, the assumption that testing samples follow the same distribution as training samples, known as the identically independent distributed (IID) assumption, often does not hold in real-world applications. When a machine learning model encounters novel testing samples that were not seen during training, it faces the out-of-distribution (OOD) generalization problem [212–217]. In brain network analysis, raw neuroimages in datasets are collected from multiple sites. Subjects from different sites may exhibit site differences due to scanner variability and differing inclusion/exclusion criteria [191]. Additionally, label inconsistencies may arise from variations in diagnostic criteria used by doctors. These site differences can cause distribution shifts, as observed in Chapter 7, and present an OOD generalization problem.

Existing Graph OOD methods [218–220] primarily focus on capturing invariant substructures from the dataset. However, brain networks are fully connected and may not maintain consistent topological structures across different subjects. Therefore, we plan to develop novel brain network OOD algorithms by (1) considering more feature information instead of relying solely on structural information in the graph and (2) utilizing brain network characteristics

such as node alignment. By addressing these gaps in OOD problems for brain network analysis, we aim to enhance the generalization ability of our models, making our algorithms more applicable across different hospitals and clinics.

9.2.3 Multi-resolution Brain Network Analysis

Traditional methods of brain network analysis often focus on networks at a single scale, disregarding the brain's inherent hierarchical organization, which comprises modular networks across various scales: individual neurons form neural circuits, neural circuits constitute functional areas (e.g., regions handling color processing), and functional areas collectively form larger functional networks (e.g., the default mode network). To address this, we plan to develop advanced machine learning models for multi-scale brain network analysis, aiming to learn more effective and discriminative brain representations for the classification of neurological diseases.

Although recent studies on multi-scale brain network analysis [102–104] have demonstrated superiority over traditional single-scale approaches, this area of research remains nascent. We have identified three critical gaps:

- The lack of publicly available multi-scale brain network benchmarks impedes progress in this crucial field.
- Existing methods are rudimentary, often failing to leverage the rich inter-scale information embedded in multi-scale brain networks.
- None of the current approaches utilize class-based information, which is essential for capturing discriminative class-specific features across different scales.

Our research will be the first systematic and extensive exploration of the full potential of multi-scale brain networks. We aim to address these gaps and pioneer the development of infrastructure for multi-scale brain network analysis.

9.2.4 Dynamic Brain Network Modeling

A significant limitation of previous GNN-based functional connectivity (FC) network analysis methods is their failure to account for the dynamic properties of FC networks, which fluctuate over time. Incorporating the dynamic features of FC networks into neuroimaging analysis is a critical direction in the field of functional neuroimaging.

Recognizing the inherent dynamism of brain activity, we aim to delve into dynamic brain network modeling. Capturing these temporal dynamics is crucial for a comprehensive understanding of neurological processes. We will explore the use of resting-state fMRI and other time-series data to gain insights into how the brain's connectivity patterns evolve over time. Dynamic brain network analysis can uncover valuable information about the brain's adaptability and response to different conditions, contributing to a more nuanced understanding of neurological disorders. This approach promises to enhance our models' ability to capture the full complexity of brain function, leading to more accurate and insightful analyses.

Besides, the integration of imaging and clinical information introduces unique challenges due to their heterogeneous nature. Brain networks represent complex node and topological features, whereas clinical data are scalar and subject-specific. This highlights the need for advanced fusion strategies that can dynamically adjust the influence of the connectome data and non-imaging information based on the task context.

Appendices

Appendix A

Supplementary for Dataset Construction and Benchmark

A.1 Extended Experimental Results

A.1.1 Results on Ordinal Regression

With the two multi-class datasets ADNI and PPMI, we study the problem of ordinal regression, in which the classes are ordered by the severity of the corresponding disease. We select the classic logistic regression model for this experiment and use the implementation in the scikit-learn library. Table A.1 reports the results of the logistic ordinal regression (LOR) versus the logistic regression (LR). The results show that LOR does not exhibit superiority to LR. One potential reason may be its sensitivity to the class imbalance problem: ADNI and PPMI have very skewed class distribution as shown in Table II in the supplementary. With the availability of the datasets, researchers could further inspect this issue and design better ordinal regression solutions to handle them.

A.1.2 Data Quality Study on ABIDE Dataset with a Graph Analysis Approach for Classification

To evaluate the quality of our data collection, we performed a recent graph-based functional analysis approach [2] to both an existing brain network dataset derived from ABIDE and our processed brain connectivity networks of the ABIDE for disease classification. The approach in Lanciano et al. [2] (model version of CS-P1) first computes two summary connectivity matrices C and P, respectively from the Control group and the Patient group. It then extracts the dense

Table A.1: Accuracy (mean±standard deviation) on logistic ordinal regression (LOR) vs logistic regression (LR) on ADNI and PPMI. The best result at each parcellation is highlighted in bold.

	ADNI					**PPMI**				
	AAL	HO	Schaefer	*k*-means	Ward	AAL	HO	Schaefer	*k*-means	Ward
LR	64.1±1.8	61.9±2.1	62.0±4.2	**58.6±2.6**	**60.4±0.8**	**56.0±7.8**	56.0±9.2	**56.5±6.8**	**60.3±7.4**	60.3±8.8
LOR	**65.6±2.2**	**62.1±3.0**	**62.8±2.2**	58.5±5.5	60.2±3.3	54.1±13.0	**57.5±8.1**	55.6±8.9	59.4±8.5	**60.8±9.1**

Table A.2: Results of CS-P1 [2] on the ABIDE dataset used in Lanciano et al. [2] and on our ABIDE dataset

	ABIDE used in Lanciano et al. [2]				**Our ABIDE**		
		Accuracy in Lanciano et al. [2]	Reproduced Accuracy			Reproduced Accuracy	
Subgroup	# Graphs	Method 1	Method 1	Method 2	# Graphs	Method 1	Method 2
Adolescents	237	72.0±7.0	71.4±3.7	64.9±7.8	259	72.2±5.5	68.3± 3.1
Children	101	86.0±7.0	83.0±9.6	69.5±8.6	96	80.1±2.6	67.1±14.0
Eyesclosed	294	71.0±3.0	68.7±4.6	60.9±3.7	312	68.6±3.5	64.8± 6.2
Male	838	63.0±1.0	63.5±2.3	60.4±5.1	873	65.3±2.1	62.7± 4.0

contrast subgraphs from C and P and uses them as features for classification. Essentially, a dense contrast subgraph refers to a subset of nodes whose induced subgraph is dense in C and sparse in P (namely contrast), or vice versa. It can be considered an Optimal Quasi-Clique (OQC) problem [221] and solved by the DENSDP approach [222]. To make the classification fairer, we list two methods of finding the contrast dense subgraph for CS-P1 [2]. Method 1 uses the whole dataset (all the subjects in both training and test sets) to extract the contrast dense subgraph. The results in Lanciano et al. [2] used this method. Method 2 uses only the training set to extract the contrast dense subgraph.

We first applied the two methods of CS-P1 to the functional networks of different subgroups of the ABIDE dataset: both the code and data were provided by the authors of Lanciano et al. [2]. The results using Method 1 (left part of Table 4) are generally comparable with the results reported in Lanciano et al. [2]: the reproduced accuracy is 3% lower than the claimed accuracy in the subgroup of "Children", and 2% lower in the subgroup of "Eyesclosed"; the results on the other two subgroups are consistent. The accuracy results using Method 2 are much lower than the results using Method 1.

We then applied CS-P1 (the same code) on our functional networks of subgroups of ABIDE dataset. Table A.2 (right) shows the result. Note that the subjects in different subgroups may overlap. Due to different data quality filtering procedures, there is a small difference in the

number of subjects in each subgroup. For both Method 1 and Method 2, the classification accuracy on our datasets is marginally better than those on the datasets provided by Lanciano et al. [2] in all subgroups except for "Children". The difference in performance is likely due to the slight difference in the number of subjects in the two datasets. The results validate that the quality of the brain networks generated by our pipeline is up to the standard of those used in other studies.

Appendix B

Supplementary for CARE

B.1 Theoretical Proofs

B.1.1 Proof Sketch of Lemma 1

Our proof follows the same flow as Lemma 1 in Kabkab et al. [171].

A parametrized class of functions with parameters in $\mathbb{R}^t$ that is computable in no more than p operations has a VC dimension which is $O(t^2p^2)$ [170]. t in GCN and CARE can be formulated as:

$$t_{GCN} = \sum_{l=0}^{d} h^l_{gcn_{in}} h^l_{gcn_{out}}; \tag{B.1}$$

$$t_{CARE} = \sum_{l=0}^{d} (h^l_{gcn_{in}} h^l_{gcn_{out}} + h^l_{gcn_{out}} + h^l_{set_{in}} h^l_{set_{out}} + h^l_{trans_{in}} h^l_{trans_{out}}). \tag{B.2}$$

By plugging in the number of multiplications $q_1(d)$ and $q_2(d)$ given by Eqs. (5.8) and (5.9), together with the above equations on the number of parameters t, into $O(t^2p^2)$, we complete the proof of Lemma 1 for both GCN and CARE.

B.1.2 Proof of Theorem 1

We compare the VC dimension upper bounds of a GCN layer and a GCN-based CARE layer under the identical number of parameters. According to Section 2.2 of Abu et al. [223], the VC dimension provides a loose generalization bound for models and can be used as a guideline for

generalization comparison - models with a lower upper bound tend to have better generalization capability. The number of parameters in a GCN layer t_1 and that in a GCN-based CARE layer t_2 are formulated as:

$$t_1 = h_{gcn_{in}} h_{gcn_{out}}, \tag{B.3}$$

$$t_2 = h_{gcn_{in}} h_{gcn_{out}} + h_{gcn_{out}} + h_{set_{in}} h_{set_{out}} + h_{trans_{in}} h_{trans_{out}}. \tag{B.4}$$

In our setting, we choose a basic hidden dimension h_1 and h_2 for GCN and CARE respectively. We set each layer to be an integer multiple of the basic hidden dimension. Thus, $t_1 = h_1^2$ and $t_2 = h_2^2 + h_2 + h_2^2 + 2h_2^2 = 4h_2^2 + h_2$, respectively.

Note that $h_{trans_{in}} = h_{set_{out}} + h_{gcn_{out}}$ as we concatenate the class representation with the subgraph representation.

Similarly, the computational complexities q_1 and q_2 can be rewritten as:

$$q_1(d) = \sum_{l=0}^{d} (nh_1^2 + n^2 h_1), \tag{B.5}$$

$$q_2(d) = \sum_{l=0}^{d} (4nh_2^2 + (2n^2 + n)h_2). \tag{B.6}$$

When $d = 1$, the complexity can be written as:

$$q_1(1) = nh_1^2 + n^2 h_1, \tag{B.7}$$

$$q_2(1) = 4nh_2^2 + (2n^2 + n)h_2. \tag{B.8}$$

Under the identical number of parameters, we let $t_1 = t_2$, and have $h_1 = \sqrt{4h_2^2 + h_2}$. Thus,

$$q_1(1) = 4nh_2^2 + nh_2 + n^2 \sqrt{4h_2^2 + h_2}. \tag{B.9}$$

The difference between $q_1(1)$ and $q_2(1)$ satisfies:

$$q_1(1) - q_2(1) = n^2(\sqrt{4h_2^2 + h_2} - 2h_2). \tag{B.10}$$

Because $\sqrt{4h_2^2 + h_2} - 2h_2 > 0$, we have:

$$q_1(1) > q_2(1). \tag{B.11}$$

According to our setting, the input and output feature map sizes of all layers is identical, which means that the 'n' in each layer's complexity equation are identical. Thus, we extend Eq. (B.11) to the full model and have:

$$q_1(d) > q_2(d). \tag{B.12}$$

With Eq. (B.12) and Lemma 1, we complete the proof of Theorem 1.

B.2 Implementation Details

The default number of graph convolutional layers in both CARE and GNN backbones is 4. We use SAGPool with a pooling ratio of 0.5 as the default subgraph selector in CARE. Notice that we did not apply any subgraph selector on GNNs that are already equipped with their own pooling methods for substructure extraction. This includes SAGPool, DiffPool, HGPSLPool and MEWISPool. The trade-off hyperparameters λ_1 and λ_2 in Eq. (5.7) are set to 1 by default. The whole network is trained in an end-to-end manner using the Adam optimizer. We use the early stopping criterion, i.e., we stop the training once there is no further improvement on the validation loss during 25 epochs. The learning rate is initialized to 10^{-4} and the maximum number of epochs is set to 1000. We set the hidden size to 146 and batch size to 20 for all models. The only exception is DiffPool when tested on the D&D dataset. Since the D&D dataset has a large number of nodes (see Table 5.2), the hidden size and batch size are set to 32 and 6 to achieve an acceptable number of parameters in DiffPool.

For TUdataset, we split it into 8:1:1 for training, validation and test. For all experiments of CARE and GNN backbones, we evaluate each model with the same random seed for 10-fold cross-validation. We use the scaffold splits for the OGBG-MOLHIV dataset and report the

average ROC-AUC with 10 random seeds. All the codes were implemented using PyTorch and Deep Graph Library packages. The experiments were conducted in a Linux server with Intel(R) Core(TM) i9-10940X CPU (3.30GHz), GeForce GTX 3090 GPU, and 125GB RAM.

B.3 Extended Experimental Results

B.3.1 Effectiveness Analysis under the same Parameter Number

CARE is proposed as a plug-and-play framework. However, in addition to directly plugging it in a GNN backbone without changing the number of model parameters in the backbone (as what we have done in experiments in the submitted version), it could also be used in a way that the resultant CARE after plug-in has a comparable number of parameters to the original GNN backbone before plug-in. This can be achieved by adjusting the number of parameters in the GNN backbone at the time when CARE is plugged in. To demonstrate this, we conduct a new experiments to match with the setting of Theorem 1. For each GNN backbone, we first set its number of parameters to 100K. For CARE, we adjust the hidden dimension of each GNN backbone to which CARE is applied such that the number of parameters of CARE is also 100K. As shown in Table B.1, CARE still outperforms its GNN backbone in 8 out of 9 cases. The results demonstrate that CARE can boost up the graph classification performance without introducing additional parameters.

Table B.1: Graph Classification Results (Average Accuracy ± Standard Deviation) under the same parameters setting. The parameter numbers of all models are 100K. Winner in each backbone/dataset pair is highlighted in **bold**.

		DD	PROTEINS	MUTAG
GraphSAGE	original	72.18 ± 2.93	74.87 ± 3.38	75.48 ± 6.11
	CARE	**72.22** ± 3.10	**75.74** ± 1.68	**76.08** ± 10.83
GCN	original	71.02 ± 3.17	73.89 ± 2.85	77.52 ± 10.81
	CARE	**71.73** ± 4.12	**74.91** ± 3.59	**79.27** ± 4.31
GIN	original	73.10 ± 2.44	72.41 ± 4.45	89.36 ± 4.71
	CARE	**73.19** ± 4.44	**70.43** ± 4.69	**89.70** ± 5.53

B.3.2 Hyperparameter Analysis

In this section, we study the sensitivity of two important hyperparameters in CARE, the trade-off parameters λ_1, λ_2 in the loss function and the number of layers. We test on three datasets using GIN as backbone for this set of experiments.

Table B.2: Results when Tuning λ_1 and λ_2.

λ_1	λ_2	D&D	PROTEINS	MUTAG
0.1	0.1	**76.32** ± 3.33	72.32 ± 4.76	85.09 ± 6.80
0.1	1	72.76 ± 4.28	72.95 ± 3.81	85.61 ± 6.41
0.1	10	73.60 ± 2.84	70.61 ± 4.04	87.22 ± 6.00
1	0.1	74.11 ± 4.38	71.87 ± 14.99	**90.47** ± 5.11
1	1	74.70 ± 3.37	72.32 ± 4.25	90.44 ± 4.58
1	10	74.45 ± 3.12	72.14 ± 4.71	89.88 ± 4.39
10	0.1	74.37 ± 3.22	**73.14** ± 3.45	89.42 ± 4.64
10	1	73.85 ± 3.75	72.49 ± 3.84	89.42 ± 5.71
10	10	73.94 ± 4.40	70.35 ± 4.73	89.97 ± 5.96

Trade-off Parameter λ_1 and λ_2. These two hyperparameters are used in the overall loss function $\mathcal{L}$ (Eq. (5.7)) to trade-off between the class loss $\mathcal{L}_{intra}$, $\mathcal{L}_{inter}$ and the classification loss $\mathcal{L}_{cls}$. We tune the value of λ_1 and λ_2 from 0.1 to 10. The results are presented in Table B.2. It shows that the choice of λ_1 and λ_2 affects the performance marginally and there doesn't exist a value that works best for all datasets. In practice, we could use the validation set to find the best value of λ_1 and λ_2.

Number of Layers. The depth of the neural network can certainly affect the model performance. We adjust the number of layers to investigate whether CARE can adapt to different depths of neural networks. We vary the number of layers from 2 to 5, and report the results in Table B.3. For each dataset, we underline the best result among all the numbers of layers tested. As shown in Table B.3, CARE consistently outperforms GIN at different number of layers, except for 4 layers on PROTEINS where the performance difference is marginal at 0.09%. The best results are achieved with 2 layers on D&D and PROTEINS and with 5 layers on MUTAG. Therefore, the number of layers should also be selected through the validation process for different datasets.

From the two hyperparameter analyses above, it is clear that while CARE can enhance the performance of various GNNs, it still requires hyperparameter tuning to identify the optimal

Table B.3: Results when Tuning Number of Layers.

Layer#	module	D&D	PROTEINS	MUTAG
2	GIN	74.11 ± 3.42	72.42 ± 2.06	89.91 ± 4.35
	CARE	**76.40** ± 2.14	**73.22** ± 2.78	**90.47** ± 5.11
3	GIN	74.53 ± 3.36	70.79 ± 5.18	87.78 ± 4.07
	CARE	**75.13** ± 3.39	**71.69** ± 4.65	**90.47** ± 5.11
4	GIN	73.10 ± 2.44	**72.41** ± 4.45	89.36 ± 4.71
	CARE	**74.70** ± 3.37	72.32 ± 4.25	**90.44** ± 4.58
5	GIN	73.93 ± 2.62	70.71 ± 4.00	91.49 ± 4.83
	CARE	**74.70** ± 3.50	**72.69** ± 3.24	**91.52** ± 5.39

configuration. This reliance on hyperparameter selection could pose a limitation to the practical application of CARE. In the future, further exploration is warranted to mitigate its sensitivity to hyperparameters, thereby enhancing the versatility of CARE.

B.4 Class Separability Metrics in Case Study

B.4.1 Silhouette Coefficient

The Silhouette of a sample x_i is defined as:

$$sil(x_i) = \frac{b^i - a^i}{max(a^i, b^i)}, \tag{B.13}$$

$$Silhouette = \underset{x_i}{AVERAGE}(\{sil(x_i)\}), \tag{B.14}$$

where a^i denotes the average distance between x_i and all other samples in the same class, and b^i denotes the smallest mean distance from x_i to all samples in any other class.

B.4.2 Separability Index

The Separability Index SI is defined as:

$$K(x_i, x_j) = \begin{cases} 1, & \text{if } y_i = y_j, y_i \in \mathcal{Y}, y_j \in \mathcal{Y} \\ 0, & \text{otherwise} \end{cases}, \tag{B.15}$$

$$x_i' = \underset{x_j \neq x_i}{\arg\min}(\|x_i - x_j\|), \tag{B.16}$$

$$SI = \frac{\sum_{x_i} K(x_i, x_i')}{m}, \tag{B.17}$$

where m is the total number of samples, x_i denotes the i-th sample, y_i denotes its corresponding class label, and $\mathcal{Y}$ denotes the set of classes. The nearest neighbour distance function $\|\cdot\|$ is assumed to utilise a suitable metric, e.g., a Manhalobis metric for symbolic data or a Euclidean metric for spatial data.

B.4.3 Hypothesis Margin

The Hypothesis Margin (HM) is defined as:

$$hm(x_i) = \frac{\|x_i - \textbf{nearmiss}(x_i)\|}{\|x_i - \textbf{nearhit}(x_i)\|}, \tag{B.18}$$

$$HM = \underset{x_i}{AVERAGE}(\{hm(x_i)\}), \tag{B.19}$$

where **nearhit**(x_i) and **nearmiss**(x_i) denote the nearest sample to x_i with the same and different label, respectively. $\|\cdot\|$ denotes the L2 distance. Note that a chosen set of features affects the margin through the distance measure.

B.4.4 Centroid Distance

The Centroid Distance (CD) is defined as:

$$c_i = AVERAGE(\{x\}_{y_x=i}), \tag{B.20}$$

$$CD = \sum_{i=1}^{|\mathcal{Y}|} \sum_{j=i}^{|\mathcal{Y}|} \|c_i - c_j\|, \tag{B.21}$$

where y_x denotes the class label of a sample x, $\mathcal{Y}$ denotes the set of classes and $\|\cdot\|$ denotes the L2 distance.

Appendix C

Supplementary for ContrastPool

C.1 Implementation Details

In ContrastPool, we adopt a GCN layer for GNN_{enc}, a GraphSAGE layer for GNN_{pool} and GNN_{emb}, and a sum pooling with a linear layer for the prediction head. For datasets with more than 2 groups (PPMI and ADNI), we use the most extreme groups to construct the contrast graph: CN and SMC vs. LMCI and AD for the ADNI dataset; NC vs. PD for PPMI. The settings of our experiments mainly follow those in [155]. We split each dataset into 8:1:1 for training, validation and test, respectively. We evaluate each model with the same random seed under 10-fold cross-validation and report the average accuracy. The hyperparameters are grid searched by Table C.1.

Table C.1: Hyperparameter settings.

batch size	4 (Taowu and Neurocon), 20 (PPMI, ADNI and ABIDE)
λ_1	{10, 1, 0.1}
λ_2	{1e-2, 1e-3, 1e-4}
pooling ratio	{0.3, 0.4, 0.5, 0.6}
L	{2, 3, 4}
learning rate	{0.02, 0.01, 0.005, 0.001}
dropout	{0, 0.1, 0.2}

The whole network is trained in an end-to-end manner using the Adam optimizer [224]. We use the early stopping criterion, i.e., we stop the training once there is no further improvement on the validation loss during 25 epochs. All the codes were implemented using PyTorch [225]

and Deep Graph Library [106] packages. All experiments were conducted on a Linux server with an Intel(R) Core(TM) i9-10940X CPU (3.30GHz), a GeForce GTX 3090 GPU, and a 125GB RAM.

C.2 Hyperparameter Analysis

In this subsection, we study the sensitivity of three important hyperparameters in ContrastPool, which are the trade-off parameters in Eq. (6.16), the pooling ratio and the number of layers. All experiments are conducted on the ABIDE dataset.

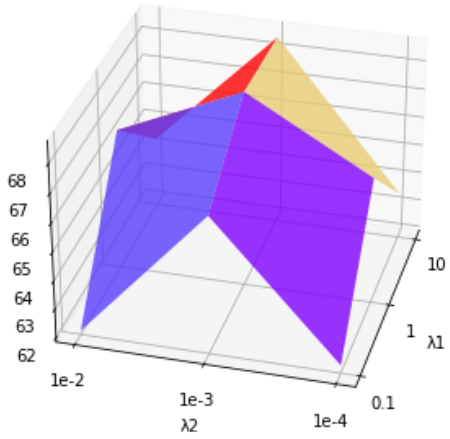


Fig. C.1: Results when tuning λ_1 and λ_2 on ABIDE.

Trade-off Parameters λ_1 and λ_2. These two hyperparameters are used in the overall loss function $\mathcal{L}$ (Eq. (6.16)) for the trade-off between the classification loss and the two entropy losses. We tune the value of λ_1 from 10 to 0.1 and λ_2 from 1e-2 to 1e-4. The results presented in Fig. C.1 show that our model performs the best when $\lambda_1 = 1$ and $\lambda_2 =$ 1e-3. The same optimal values of λ_1 and λ_2 are found on other datasets.

Pooling Ratio. The number of nodes $m^{(l)}$ in the output graph of each ContrastPool layer is controlled by the pooling ratio. A smaller pooling ratio would lead to fewer node clusters in each ContrastPool layer. Herein, we tune it from 0.3 to 0.6. As shown in Fig. C.2, the best choice of pooling ratio is 0.4. The best pooling ratio on different datasets varies slightly within the range of [0.4, 0.5].

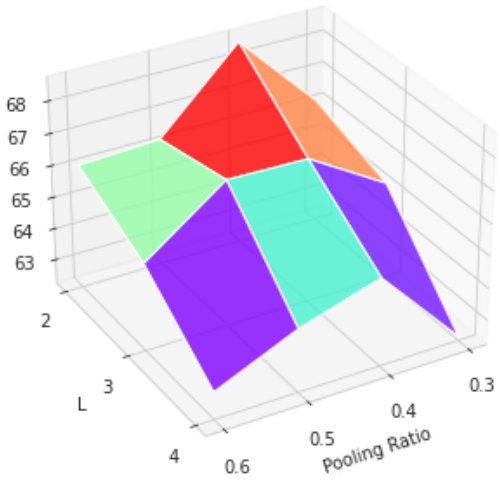


Fig. C.2: Results when tuning pooling ratio and the number of layers on ABIDE.

Number of Layers. The depth of the neural network can undoubtedly affect the model performance. We vary the number of layers L in ContrastPool from 2 to 4, and report the results in Fig. C.2. ContrastPool achieves the best performance when we set L to 2. This indicates that by leveraging the contrast graph, our ContrastPool requires fewer layers to obtain good representations, while most other GNN baselines need to be deeper (e.g., 4 layers) to achieve best performance. The same conclusion of the optimal L can be drawn on other datasets.

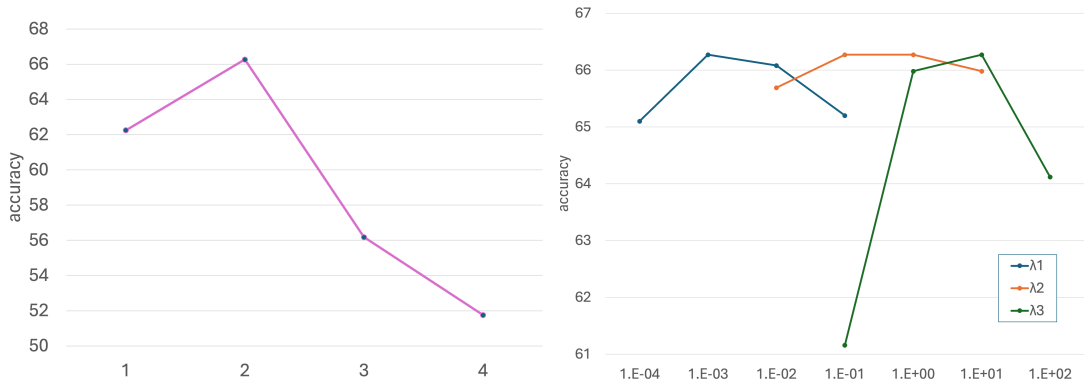
Appendix D

Supplementary for Contrasformer

D.1 Hyperparameter Analysis

In this subsection, we study the sensitivity of four important hyperparameters in Contrasformer, which are the number of layers L and the trade-off parameters in Eq. (7.10). All experiments are conducted on ABIDE dataset.

Number of Layers. The depth of the neural network can undoubtedly affect the model performance. We vary the number of layers L in Contrasformer from 1 to 4, and report the results in Fig. D.1(a). Contrasformer achieves the best performance when we set L to 2. The same conclusion of the optimal L can be drawn on other datasets. It can be found that too many Contrasformer layers will inevitably introduce noise to the model instead of extracting the truly discriminative information.



(a) Tuning number of layers.

(b) Tuning λ_1, λ_2 and λ_3.

Fig. D.1: The performance of Contrasformer on ABIDE with different hyperparameters.

Trade-off Parameters λ_1, λ_2 and λ_3. These hyperparameters are used in the overall loss function $\mathcal{L}_{total}$ (Eq. 7.10) for the trade-off between the classification loss and the three auxiliary losses. We tune the value of λ_1 from 10^{-1} to 10^{-4}, λ_2 from 10.0 to 10^{-2} and λ_3 from 100.0 to 10^{-1}. All these hyperparameters are tuned independently with other hyperparameters fixed to the best value. The classification results presented in Fig. D.1(b) show that our model performs the best when $\lambda_1 = 10^{-3}$, $\lambda_2 = 10^{-1}$ and $\lambda_3 = 10.0$. It can be found that comparing with λ_1 and λ_2, λ_3 has the greatest impact on the performance.

Appendix E

Supplementary for AIDFusion

E.1 Difference between Multi-atlas and Multi-template Methods

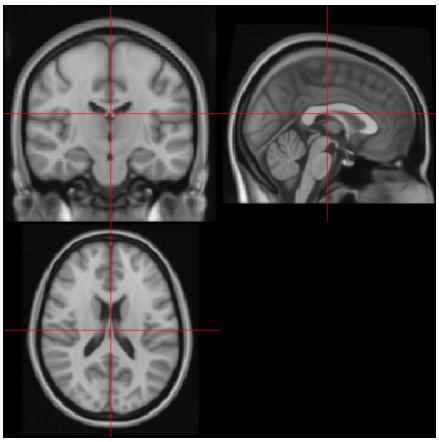


Fig. E.1: MNI template T1-w image.

In brain MRI analysis, the concepts of template and atlas can sometimes be confused. The atlas in our paper refers to a detailed map of brain structures [27], often derived from anatomical, functional, and histological data, and includes labels for different brain regions based on specific criteria. A template, on the other hand, is a standard reference image that serves as a common coordinate system for comparing different brain images [226]. It is usually created by averaging brain images from a group of subjects, providing a standardized space for registering or aligning individual brain images. This allows for comparison and combination

of data across different subjects or groups. Fig. E.1 provides a brain template. Although some existing works [227–230] are named multi-atlas, they are more akin to multi-template methods. Instead of registering brain images to different spaces in multi-template methods, the multi-atlas methods discussed in our paper segment brain images in the common space to define ROIs differently.

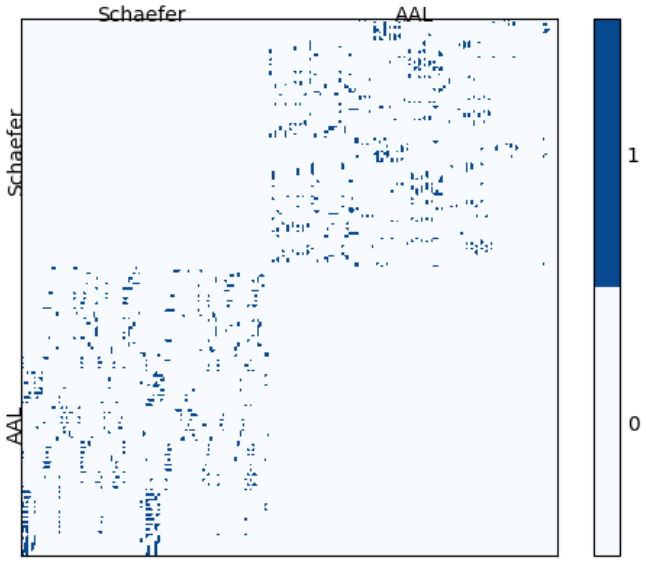


Fig. E.2: The adjacency matrix for inter-atlas message-passing.

E.2 An example of the Adjacency Matrix for Inter-Atlas Message-Passing

As shown in Fig. E.2, we generated an adjacency matrix for Schaefer100 and AAL116 by setting $k = 5$. The connections between spatial neighborhoods across atlases are constructed. Note that node v in atlas 1 is one of the nearest k neighbors of node u in atlas 2 does not mean u is one of the nearest k neighbors of v, thus the adjacency matrix is asymmetry.

E.3 Implementation Details

The settings of our experiments mainly follow those in Dwivedi et al. [155]. We split each dataset into 8:1:1 for training, validation and test, respectively. We evaluate each model with

the same random seed under 10-fold cross-validation and report the average accuracy. The whole network is trained in an end-to-end manner using the Adam optimizer [224]. We use the early stopping criterion, i.e., we halve the learning rate when there is no further improvement on the validation loss during 25 epochs and stop the training once the learning rate is smaller than the minimum rate we set. All the codes were implemented using PyTorch [225] and Deep Graph Library [106] packages. All experiments were conducted on a Linux server with an AMD Ryzen Threadripper PRO 5995WX 64-Cores and an NVIDIA GeForce RTX 4090. The version for the software we used in AIDFusion is listed in Table E.1.

Table E.1: The software dependency of AIDFusion.

Dependency	Version
Python	3.10.13
cudatoolkit	12.2
pytorch	2.2.1+cu121
DGL	2.1.0+cu121
scikit-learn	1.4.1.post1
numpy	1.26.4
matplotlib	3.8.3
nilearn	0.10.4

E.4 Hyperparameter Analysis

In this section, we study the sensitivity of four trade-off hyperparameters in Eq. (8.13). All experiments are conducted on the ADNI dataset. We tune the value of λ_1 from 1e0 to 1e2, λ_2 from 1e0 to 1e2, λ_3 from 1e-6 to 1e-4 and λ_4 from 1e-1 to 1e1. The results presented in Table E.2 show that our model performs the best when λ_1 = 1e1, λ_2 = 1e1, λ_3 = 1e-5 and λ_4 = 1e0. We can exhibit that these trade-off hyperparameters in the loss function will marginally affect the model performance on ADNI (less than 1%), which demonstrates the stability of AIDFusion.

Table E.2: The hyperparameter sensitivity analysis for AIDFusion on ADNI dataset.

λ_1	λ_2	λ_3	λ_4	acc ± std
1e0	1e1	1e-5	1e0	66.97 ± 1.95
1e1	1e0	1e-5	1e0	66.44 ± 2.88
1e1	1e1	1e-6	1e0	67.04 ± 2.20
1e1	1e1	1e-5	1e-1	66.82 ± 1.98
1e1	1e1	1e-5	1e0	**67.57** ± 2.04
1e1	1e1	1e-5	1e1	66.82 ± 2.64
1e1	1e1	1e-4	1e0	67.04 ± 2.21
1e1	1e2	1e-5	1e0	66.89 ± 2.10
1e2	1e1	1e-5	1e0	66.21 ± 2.34

List of Publications

Journal Articles

- **Jiaxing Xu**, Qingtian Bian, Xinhang Li, Aihu Zhang, Yiping Ke, Miao Qiao, Wei Zhang, Wei Khang Jeremy Sim, Balázs Gulyás. "Contrastive Graph Pooling for Explainable Classification of Brain Networks." in *IEEE Transactions on Medical Imaging*, 2024.
- **Jiaxing Xu**, Jinjie Ni, and Yiping Ke. "A Class-Aware Representation Refinement Framework for Graph Classification." in *Information Sciences*, 2024.
- Tieyiing Li, Lingdu Kong, Xiaochun Yang, Bin Wang, **Jiaxing Xu**. "Bridging Modalities: A Survey of Cross-Modal Image-Text Retrieval." in *Chinese Journal of Information Fusion*, 2024.

Conference Proceedings

- **Jiaxing Xu**, Kai He, Mengcheng Lan, Qingtian Bian, Wei Li, Tieying Li, Yiping Ke, Miao Qiao. "Contrasformer: A Brain Network Contrastive Transformer for Neurodegenerative Condition Identification." in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024.
- Tieying Li, Xiaochun Yang, Yiping Ke, Bin Wang, Yinan Liu, **Jiaxing Xu**. "Alleviating the Inconsistency of Multimodal Data in Cross-Modal Retrieval." in *Proceeding of the IEEE International Conference of Data Engineering*, 2024.
- **Jiaxing Xu***, Aihu Zhang*, Qingtian Bian, Yiping Ke. "Union Subgraph Neural Network." in *Proceedings of the 38th Annual AAAI Conference on Artificial Intelligence*, 2024.

- **Jiaxing Xu***, Yunhan Yang*, David Tse Jung Huang*, Sophi Shilpa Gururajapathy*, Yiping Ke, Miao Qiao, Alan Wang, Haribalan Kumar, Josh McGeown, and Eryn Kwon. "Data-Driven Network Neuroscience: On Data Collection and Benchmark." in *Proceedings of the 37th Conference on Neural Information Processing Systems*, 2023.

- Mengcheng Lan, Xinjiang Wang, Yiping Ke, **Jiaxing Xu**, Litong Feng, and Wayne Zhang. "SmooSeg: Smoothness Prior for Unsupervised Semantic Segmentation." in *Proceedings of the 37th Conference on Neural Information Processing Systems*, 2023.

- Qingtian Bian, **Jiaxing Xu**, Hui Fang, Yiping Ke. "CPMR: Context-Aware Incremental Sequential Recommendation with Pseudo-Multi-Task Learning." in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023.

- Xinhang Li, Xiangyu Zhao, **Jiaxing Xu**, Yong Zhang, Chunxiao Xing. "IMF: Interactive Multimodal Fusion Model for Link Prediction." in *Proceedings of the ACM Web Conference*, 2023.

Preprint

- **Jiaxing Xu**, Mengcheng Lan, Xia Dong, Kai He, Wei Zhang, Qingtian Bian, Yiping Ke. "Multi-Atlas Brain Network Classification through Consistency Distillation and Complementary Information Fusion". *Under Review*.

- Miao Qiao, Yunhan Yang, **Jiaxing Xu**, Yiping Ke, Jing Sun. "On Data-driven Brain Network Analysis: Is the Spatial Information Important?" *Under Review*.

- Qingtian Bian, Tieying Le, Marcus Vinícius Sousa Leite de Carvalho, **Jiaxing Xu**, Hui Fang and Yiping Ke. "Mitigating Domain Convergence of Mutual Transfer in Cross-Domain Sequential Recommendation". *Under Review*.

- Han Lei, **Jiaxing Xu**, Jinjie Ni, Yiping Ke. "Rethinking the Message Passing for Graph-Level Representation Learning in a Category-Based View". *Under Review*.

- Han Lei, **Jiaxing Xu**, Xia Dong, Yiping Ke. "Divergent Paths: Separating Homophilic and Heterophilic Learning for Enhanced Graph-level Representations". *Under Review*.

- Wei Li, **Jiaxing Xu**, Xia Dong, Yiping Ke. "Bridge Breaking for Graph Neural Networks". *Under Review*.

Bibliography

[1] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008. xvi, 53, 72

[2] T. Lanciano, F. Bonchi, and A. Gionis, "Explainable classification of brain networks via contrast subgraphs," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3308–3318. xx, 2, 3, 6, 10, 11, 14, 22, 31, 32, 33, 62, 89, 112, 113, 114

[3] S. Chen, Z. He, X. Han, X. He, R. Li, H. Zhu, D. Zhao, C. Dai, Y. Zhang, Z. Lu *et al.*, "How big data and high-performance computing drive brain science," *Genomics, proteomics & bioinformatics*, vol. 17, no. 4, pp. 381–392, 2019. 1

[4] M. D. Fox and M. Greicius, "Clinical applications of resting state functional connectivity," *Frontiers in systems neuroscience*, vol. 4, p. 19, 2010. 1, 22, 24

[5] R. A. Poldrack, Y. O. Halchenko, and S. J. Hanson, "Decoding the large-scale structure of brain function by classifying mental states across individuals," *Psychological science*, vol. 20, no. 11, pp. 1364–1372, 2009. 1, 88

[6] J. Xu, Y. Yang, D. T. J. Huang, S. S. Gururajapathy, Y. Ke, M. Qiao, A. Wang, H. Kumar, J. McGeown, and E. Kwon, "Data-driven network neuroscience: On data collection and benchmark," in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 2, 22, 89, 99

[7] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *Advances in neural information processing systems*, vol. 28, 2015. 2

[8] H. Dai, B. Dai, and L. Song, "Discriminative embeddings of latent variable models for structured data," in *International conference on machine learning*. PMLR, 2016, pp. 2702–2711. 2

[9] D. Masters, J. Dean, K. Klaser, Z. Li, S. Maddrell-Mander, A. Sanders, H. Helal, D. Beker, L. Rampášek, and D. Beaini, "Gps++: An optimised hybrid mpnn/transformer for molecular property prediction," *arXiv preprint arXiv:2212.02229*, 2022. 2

[10] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," *Advances in neural information processing systems*, vol. 31, 2018. 2, 13, 47, 60, 61, 62, 94

[11] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426. 2

[12] W. Shiao, Z. Guo, T. Zhao, E. E. Papalexakis, Y. Liu, and N. Shah, "Link prediction with non-contrastive learning," *arXiv preprint arXiv:2211.14394*, 2022. 2

[13] H. Peng, H. Wang, B. Du, M. Z. A. Bhuiyan, H. Ma, J. Liu, L. Wang, Z. Yang, L. Du, S. Wang *et al.*, "Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting," *Information Sciences*, vol. 521, pp. 277–290, 2020. 2

[14] A. Derrow-Pinion, J. She, D. Wong, O. Lange, T. Hester, L. Perez, M. Nunkesser, S. Lee, X. Guo, B. Wiltshire *et al.*, "Eta prediction with graph neural networks in google maps," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3767–3776. 2

[15] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, and L. Petersson, "Graph-based deep learning for medical diagnosis and analysis: past, present and future," *Sensors*, vol. 21, no. 14, p. 4758, 2021. 2

[16] J. Wang, A. Ma, Y. Chang, J. Gong, Y. Jiang, R. Qi, C. Wang, H. Fu, Q. Ma, and D. Xu, "scgnn is a novel graph neural network framework for single-cell rna-seq analyses," *Nature communications*, vol. 12, no. 1, p. 1882, 2021. 2

[17] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," *arXiv preprint arXiv:2009.03509*, 2020. 2

[18] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," *Advances in neural information processing systems*, vol. 31, 2018. 2

[19] S. Suresh, M. Shrivastava, A. Mukherjee, J. Neville, and P. Li, "Expressive and efficient representation learning for ranking links in temporal graphs," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 567–577. 2

[20] J. Kawahara, C. J. Brown, S. P. Miller, B. G. Booth, V. Chau, R. E. Grunau, J. G. Zwicker, and G. Hamarneh, "Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment," *NeuroImage*, vol. 146, pp. 1038–1049, 2017. 2, 14, 23, 32, 62, 89, 97

[21] F. Errica, M. Podda, D. Bacciu, and A. Micheli, "A fair comparison of graph neural networks for graph classification," *arXiv preprint arXiv:1912.09893*, 2019. 3, 22

[22] Y. Liu, Y. Liu, and K. Chan, "Ordinal regression via manifold learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, pp. 398–403, Aug. 2011. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/7937 3, 22

[23] S. I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, and D. Rueckert, "Distance metric learning using graph convolutional networks: Application to functional brain networks," in *Medical Image Computing and Computer Assisted Intervention-MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part I 20*. Springer, 2017, pp. 469–477. 3, 14, 23, 55

[24] X. Li, Y. Zhou, N. C. Dvornek, M. Zhang, J. Zhuang, P. Ventola, and J. S. Duncan, "Pooling regularized graph neural network for fmri biomarker analysis," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VII 23*. Springer, 2020, pp. 625–635. 3, 14, 23, 55, 62, 71, 73

[25] X. Li, Y. Zhou, N. Dvornek, M. Zhang, S. Gao, J. Zhuang, D. Scheinost, L. H. Staib, P. Ventola, and J. S. Duncan, "Braingnn: Interpretable brain graph neural network for fmri analysis," *Medical Image Analysis*, vol. 74, p. 102233, 2021. 3, 14, 23, 55, 62, 63, 71, 73

[26] Y. Yan, J. Zhu, M. Duda, E. Solarz, C. Sripada, and D. Koutra, "Groupinn: Grouping-based interpretable neural network for classification of limited, noisy brain data," in *proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 772–782. 4, 14, 55

[27] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot, "Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain," *Neuroimage*, vol. 15, no. 1, pp. 273–289, 2002. 5, 23, 30, 89, 127

[28] N. Makris, J. M. Goldstein, D. Kennedy, S. M. Hodge, V. S. Caviness, S. V. Faraone, M. T. Tsuang, and L. J. Seidman, "Decreased volume of left and total anterior insular lobule in schizophrenia," *Schizophrenia research*, vol. 83, no. 2-3, pp. 155–171, 2006. 5, 23, 30, 89, 99

[29] A. Schaefer, R. Kong, E. M. Gordon, T. O. Laumann, X.-N. Zuo, A. J. Holmes, S. B. Eickhoff, and B. T. Yeo, "Local-global parcellation of the human cerebral cortex from intrinsic functional connectivity mri," *Cerebral cortex*, vol. 28, no. 9, pp. 3095–3114, 2018. 5, 30, 89

[30] Z. Long, J. Li, H. Liao, L. Deng, Y. Du, J. Fan, X. Li, J. Miao, S. Qiu, C. Long *et al.*, "A multi-modal and multi-atlas integrated framework for identification of mild cognitive impairment," *Brain Sciences*, vol. 12, no. 6, p. 751, 2022. 5, 89

[31] Y. Chu, G. Wang, L. Cao, L. Qiao, and M. Liu, "Multi-scale graph representation learning for autism identification with functional mri," *Frontiers in Neuroinformatics*, vol. 15, p. 802305, 2022. 5, 15, 90, 93, 97

[32] L. Mahler, Q. Wang, J. Steiglechner, F. Birk, S. Heczko, K. Scheffler, and G. Lohmann, "Pretraining is all you need: A multi-atlas enhanced transformer framework for autism spectrum disorder classification," in *International Workshop on Machine Learning in Clinical Neuroimaging*. Springer, 2023, pp. 123–132. 5, 15, 90, 97

[33] D.-J. Lee, D.-H. Shin, Y.-H. Son, J.-W. Han, J.-H. Oh, D.-H. Kim, J.-H. Jeong, and T.-E. Kam, "Spectral graph neural network-based multi-atlas brain network fusion for major depressive disorder diagnosis," *IEEE Journal of Biomedical and Health Informatics*, 2024. 5, 15, 90, 93, 97

[34] C. Craddock, Y. Benhajali, C. Chu, F. Chouinard, A. Evans, A. Jakab, B. S. Khundrakpam, J. D. Lewis, Q. Li, M. Milham *et al.*, "The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives," *Frontiers in Neuroinformatics*, vol. 7, 2013. 10, 23, 72

[35] P. Bellec, S. Lavoie-Courchesne, P. Dickinson, J. P. Lerch, A. P. Zijdenbos, and A. C. Evans, "The pipeline system for octave and matlab (PSOM): a lightweight scripting framework and execution engine for scientific workflows," *Frontiers Neuroinformatics*, vol. 6, p. 7, 2012. [Online]. Available: https://doi.org/10.3389/fninf.2012.00007 10

[36] O. Esteban, C. J. Markiewicz, R. W. Blair, C. A. Moodie, A. I. Isik, A. Erramuzpe, J. D. Kent, M. Goncalves, E. DuPre, M. Snyder *et al.*, "fmriprep: a robust preprocessing pipeline for functional mri," *Nature methods*, vol. 16, no. 1, pp. 111–116, 2019. 10, 29

[37] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "Tu-dataset: A collection of benchmark datasets for learning with graphs," *arXiv preprint arXiv:2007.08663*, 2020. 10

[38] A. Fornito, A. Zalesky, and E. Bullmore, *Fundamentals of brain network analysis*. Academic Press, 2016. 11

[39] I. Bilgen, G. Guvercin, and I. Rekik, "Machine learning methods for brain network classification: Application to autism diagnosis using cortical morphological networks," *Journal of neuroscience methods*, vol. 343, p. 108799, 2020. 11, 32, 33

[40] Z. Rakhimberdina, X. Liu, and T. Murata, "Population graph-based multi-model ensemble method for diagnosing autism spectrum disorder," *Sensors*, vol. 20, no. 21, p. 6001, 2020. 11, 33

[41] J. Liu, Y. Sheng, W. Lan, R. Guo, Y. Wang, and J. Wang, "Improved asd classification using dynamic functional connectivity and multi-task feature selection," *Pattern Recognition Letters*, vol. 138, pp. 82–87, 2020. 11, 31, 33

[42] M. Ingalhalikar, S. Shinde, A. Karmarkar, A. Rajan, D. Rangaprakash, and G. Deshpande, "Functional connectivity-based prediction of autism on site harmonized abide dataset," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 12, pp. 3628–3637, 2021. 11, 32, 33

[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017. 11, 32, 43, 46, 62, 72, 73

[44] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021. 11

[45] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "Gram: graph-based attention model for healthcare representation learning," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 787–795. 11

[46] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. Gao, "Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 1903–1911. 11

[47] J. B. Lee, R. Rossi, and X. Kong, "Graph classification using structural attention," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1666–1674. 11

[48] G. Wang, R. Ying, J. Huang, and J. Leskovec, "Multi-hop attention graph neural network," *arXiv preprint arXiv:2009.14332*, 2020. 11

[49] T. He, Y. S. Ong, and L. Bai, "Learning conjoint attentions for graph neural nets," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2641–2653, 2021. 11

[50] W. Zhang, Z. Yin, Z. Sheng, Y. Li, W. Ouyang, X. Li, Y. Tao, Z. Yang, and B. Cui, "Graph attention multi-layer perceptron," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4560–4570. 11

[51] L. Lin, E. Blaser, and H. Wang, "Graph embedding with hierarchical attentive membership," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 582–590. 11

[52] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670*, 2018. 12

[53] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738. 12

[54] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607. 12, 80

[55] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5812–5823, 2020. 12, 39, 79, 80, 95

[56] Q. Sun, J. Li, H. Peng, J. Wu, Y. Ning, P. S. Yu, and L. He, "Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism," in *Proceedings of the Web Conference 2021*, 2021, pp. 2081–2091. 12

[57] Y. Yin, Q. Wang, S. Huang, H. Xiong, and X. Zhang, "Autogcl: Automated graph contrastive learning via learnable view generators," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 8, 2022, pp. 8892–8900. 12

[58] J. Chen and G. Kou, "Attribute and structure preserving graph contrastive learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7024–7032. 12

[59] C. Liu, Y. Zhan, J. Wu, C. Li, B. Du, W. Hu, T. Liu, and D. Tao, "Graph pooling for graph neural networks: Progress, challenges, and opportunities," *arXiv preprint arXiv:2204.07321*, 2022. 12

[60] H. Gao and S. Ji, "Graph u-nets," in *international conference on machine learning*. PMLR, 2019, pp. 2083–2092. 13

[61] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3734–3743. 13, 41, 46, 49, 62

[62] Z. Zhang, J. Bu, M. Ester, J. Zhang, C. Yao, Z. Yu, and C. Wang, "Hierarchical graph pooling with structure learning," *arXiv preprint arXiv:1911.05954*, 2019. 13, 47, 49, 62

[63] L. Zhang, X. Wang, H. Li, G. Zhu, P. Shen, P. Li, X. Lu, S. A. A. Shah, and M. Bennamoun, "Structure-feature based graph self-adaptive pooling," in *Proceedings of The Web Conference 2020*, 2020, pp. 3098–3104. 13

[64] H. Gao, Y. Liu, and S. Ji, "Topology-aware graph pooling networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4512–4518, 2021. 13

[65] M. Li, S. Chen, Y. Zhang, and I. W. Tsang, "Graph cross networks with vertex infomax pooling," *arXiv preprint arXiv:2010.01804*, 2020. 13, 47

[66] J. Qin, L. Liu, H. Shen, and D. Hu, "Uniform pooling for graph networks," *Applied Sciences*, vol. 10, no. 18, p. 6287, 2020. 13

[67] X. Gao, W. Dai, C. Li, H. Xiong, and P. Frossard, "ipool—information-based pooling in hierarchical graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 5032–5044, 2021. 13

[68] H. Yuan and S. Ji, "Structpool: Structured graph pooling via conditional random fields," in *Proceedings of the 8th international conference on learning representations*, 2020. 13

[69] E. Noutahi, D. Beaini, J. Horwood, S. Giguère, and P. Tossou, "Towards interpretable sparse graph representation learning with laplacian pooling," *arXiv preprint arXiv:1905.11577*, 2019. 13

[70] F. M. Bianchi, D. Grattarola, and C. Alippi, "Spectral clustering with graph neural networks for graph pooling," in *International conference on machine learning*. PMLR, 2020, pp. 874–883. 13

[71] A. H. K. Ahmadi, *Memory-based graph networks*. University of Toronto (Canada), 2020. 13

[72] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 13, 19, 58, 72, 75, 76, 78, 92, 97

[73] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," *arXiv preprint arXiv:2012.09699*, 2020. 14

[74] D. Kreuzer, D. Beaini, W. Hamilton, V. Létourneau, and P. Tossou, "Rethinking graph transformers with spectral attention," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 618–21 629, 2021. 14

[75] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform badly for graph representation?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 877–28 888, 2021. 14, 72, 77, 91

[76] L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini, "Recipe for a general, powerful, scalable graph transformer," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14 501–14 515, 2022. 14, 72

[77] X. Kan, W. Dai, H. Cui, Z. Zhang, Y. Guo, and C. Yang, "Brain network transformer," *arXiv preprint arXiv:2210.06681*, 2022. 14, 15, 23, 62

[78] V. P. Dwivedi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson, "Graph neural networks with learnable structural and positional representations," *arXiv preprint arXiv:2110.07875*, 2021. 14

[79] X. Li, N. C. Dvornek, Y. Zhou, J. Zhuang, P. Ventola, and J. S. Duncan, "Graph neural network for interpreting task-fmri biomarkers," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part V 22*. Springer, 2019, pp. 485–493. 14, 23, 62

[80] M. Xu, D. L. Sanz, P. Garces, F. Maestu, Q. Li, and D. Pantazis, "A graph gaussian embedding method for predicting alzheimer's disease progression with meg brain networks," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 5, pp. 1579–1588, 2021. 14, 62, 97

[81] H. Zhang, R. Song, L. Wang, L. Zhang, D. Wang, C. Wang, and W. Zhang, "Classification of brain disorders in rs-fmri via local-to-global graph neural networks," *IEEE Transactions on Medical Imaging*, 2022. 15, 23, 71, 73

[82] B.-H. Kim, J. C. Ye, and J.-J. Kim, "Learning dynamic graph representation of brain connectome with spatio-temporal attention," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4314–4327, 2021. 15

[83] Y. Yu, X. Kan, H. Cui, R. Xu, Y. Zheng, X. Song, Y. Zhu, K. Zhang, R. Nabi, Y. Guo *et al.*, "Learning task-aware effective brain connectivity for fmri analysis with graph neural networks," *arXiv preprint arXiv:2211.00261*, 2022. 15

[84] J. Liu, W. Cui, Y. Chen, Y. Ma, Q. Dong, R. Cai, Y. Li, and B. Hu, "Deep fusion of multi-template using spatio-temporal weighted multi-hypergraph convolutional networks for brain disease analysis," *IEEE Transactions on Medical Imaging*, 2023. 15

[85] F. Huang, E.-L. Tan, P. Yang, S. Huang, L. Ou-Yang, J. Cao, T. Wang, and B. Lei, "Self-weighted adaptive structure learning for asd diagnosis via multi-template multi-center representation," *Medical image analysis*, vol. 63, p. 101662, 2020. 15

[86] A. Shehzad, S. Yu, D. Zhang, S. Abid, X. Cheng, J. Zhou, and F. Xia, "Braingt: Multifunctional brain graph transformer for brain disorder diagnosis," Aug. 2024. [Online]. Available: http://medrxiv.org/lookup/doi/10.1101/2024.08.30.24312819 15

[87] W. Wang, X. Hu, L. Xiao, and Y.-P. Wang, "Adaptive multiview community-preserved graph convolutional network for multiatlas-based functional connectivity analysis," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Seoul, Korea, Republic of: IEEE, Apr. 2024, p. 2056–2060. [Online]. Available: https://ieeexplore.ieee.org/document/10448137/ 15

[88] Y. Ma, W. Cui, J. Liu, Y. Guo, H. Chen, and Y. Li, "A multi-graph cross-attention-based region-aware feature fusion network using multi-template for brain disorder diagnosis," *IEEE Transactions on Medical Imaging*, vol. 43, no. 3, p. 1045–1059, Mar. 2024. 15

[89] W. Wang and L. Xiao, *Consistency Guided Multiview Hypergraph Embedding Learning with Multiatlas-Based Functional Connectivity Networks Using Resting-State fMRI*, ser. Lecture Notes in Computer Science. Singapore: Springer Nature Singapore, 2024, vol. 14433, p. 170–181. [Online]. Available: https://link.springer.com/10.1007/978-981-99-8546-3_14 15

[90] W. Wang, L. Xiao, G. Qu, V. D. Calhoun, Y.-P. Wang, and X. Sun, "Multiview hyperedge-aware hypergraph embedding learning for multisite, multiatlas fmri based functional connectivity network analysis," *Medical Image Analysis*, vol. 94, p. 103144, May 2024. 16

[91] X. Liu, M. R. Hasan, T. Gedeon, and M. Z. Hossain, "Made-for-asd: A multi-atlas deep ensemble network for diagnosing autism spectrum disorder," *Computers in Biology and Medicine*, vol. 182, p. 109083, Nov. 2024. 16

[92] H. Yang, X. Li, Y. Wu, S. Li, S. Lu, J. S. Duncan, J. C. Gee, and S. Gu, "Interpretable multimodality embedding of cerebral cortex using attention graph network for identifying bipolar disorder," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*. Springer, 2019, pp. 799–807. 16, 23

[93] T. Zhou, M. Liu, K.-H. Thung, and D. Shen, "Latent representation learning for alzheimer's disease diagnosis with incomplete multi-modality neuroimaging and genetic data," *IEEE transactions on medical imaging*, vol. 38, no. 10, pp. 2411–2422, 2019. 16, 107

[94] T. Zhou, K.-H. Thung, M. Liu, F. Shi, C. Zhang, and D. Shen, "Multi-modal latent space inducing ensemble svm classifier for early dementia diagnosis with neuroimaging data," *Medical image analysis*, vol. 60, p. 101630, 2020. 16, 107

[95] V. Le Du, C. Presigny, A. Bouzigues, V. Godefroy, B. Batrancourt, R. Levy, F. D. V. Fallani, and R. Migliaccio, "Multi-atlas multilayer brain networks, a new multimodal approach to neurodegenerative disease," in *2021 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART)*. IEEE, 2021, pp. 1–5. 16, 107

[96] D. Yao, J. Sui, M. Wang, E. Yang, Y. Jiaerken, N. Luo, P.-T. Yap, M. Liu, and D. Shen, "A mutual multi-scale triplet graph convolutional network for classification of brain disorders using functional or structural connectivity," *IEEE transactions on medical imaging*, vol. 40, no. 4, pp. 1279–1289, 2021. 16, 107

[97] Q. Zhu, H. Wang, B. Xu, Z. Zhang, W. Shao, and D. Zhang, "Multimodal triplet attention network for brain disease diagnosis," *IEEE Transactions on Medical Imaging*, vol. 41, no. 12, pp. 3884–3894, 2022. 16, 107

[98] X. Wei, K. Zhao, Y. Jiao, N. B. Carlisle, H. Xie, G. A. Fonzo, and Y. Zhang, "Multimodal cross-domain self-supervised pre-training for fmri and eeg fusion," *Neural Networks*, p. 107066, 2024. 16

[99] Y. Yang, C. Ye, G. Cai, K. Song, J. Zhang, Y. Xiang, and T. Ma, "Hypercomplex graph neural network: Towards deep intersection of multi-modal brain networks," *IEEE Journal of Biomedical and Health Informatics*, 2024. 16

[100] Z. Li, H. Li, A. L. Ralescu, J. R. Dillman, M. Altaye, K. M. Cecil, N. A. Parikh, and L. He, "Joint self-supervised and supervised contrastive learning for multimodal mri data: Towards predicting abnormal neurodevelopment," *Artificial Intelligence in Medicine*, vol. 157, p. 102993, 2024. 16

[101] S. Li and R. Zhang, "A novel interactive deep cascade spectral graph convolutional network with multi-relational graphs for disease prediction," *Neural Networks*, vol. 175, p. 106285, 2024. 16

[102] M. Liu, H. Zhang, F. Shi, and D. Shen, "Building dynamic hierarchical brain networks and capturing transient meta-states for early mild cognitive impairment diagnosis," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VII 24*. Springer, 2021, pp. 574–583. 16, 109

[103] ——, "Hierarchical graph convolutional network built by multiscale atlases for brain disorder diagnosis using functional connectivity," *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 16, 109

[104] X. Wen, Q. Cao, B. Jing, and D. Zhang, "Multi-scale fc-based multi-order gcn: A novel model for predicting individual behavior from fmri," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2024. 16, 109

[105] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2018. 19, 32, 43, 46, 62, 73

[106] M. Wang, L. Yu, D. Zheng, Q. Gan, Y. Gai, Z. Ye, M. Li, J. Zhou, Q. Huang, C. Ma *et al.*, "Deep graph library: Towards efficient and scalable deep learning on graphs." 2019. 19, 123, 129

[107] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint arXiv:1903.02428*, 2019. 19

[108] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 19

[109] A. Venkataraman, D. Y.-J. Yang, K. A. Pelphrey, and J. S. Duncan, "Bayesian community detection in the space of group-level functional differences," *IEEE transactions on medical imaging*, vol. 35, no. 8, pp. 1866–1882, 2016. 23

[110] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, K. Ugurbil, W.-M. H. Consortium *et al.*, "The wu-minn human connectome project: an overview," *Neuroimage*, vol. 80, pp. 62–79, 2013. 23

[111] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman *et al.*, "An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest," *Neuroimage*, vol. 31, no. 3, pp. 968–980, 2006. 23

[112] A. S. Greene, S. Gao, D. Scheinost, and R. T. Constable, "Task-induced brain state manipulation improves prediction of individual traits," *Nature communications*, vol. 9, no. 1, p. 2807, 2018. 23

[113] C. Destrieux, B. Fischl, A. Dale, and E. Halgren, "Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature," *Neuroimage*, vol. 53, no. 1, pp. 1–15, 2010. 23

[114] A. Daducci, S. Gerhard, A. Griffa, A. Lemkaddem, L. Cammoun, X. Gigandet, R. Meuli, P. Hagmann, and J.-P. Thiran, "The connectome mapper: an open-source processing pipeline to map connectomes with mri," *PloS one*, vol. 7, no. 12, p. e48121, 2012. 23

[115] B. G. Booth, S. P. Miller, C. J. Brown, K. J. Poskitt, V. Chau, R. E. Grunau, A. R. Synnes, and G. Hamarneh, "Steam—statistical template estimation for abnormality mapping: A personalized dti analysis technique with applications to the screening of preterm infants," *NeuroImage*, vol. 125, pp. 705–723, 2016. 23

[116] F. Shi, P.-T. Yap, G. Wu, H. Jia, J. H. Gilmore, W. Lin, and D. Shen, "Infant brain atlases from neonates to 1-and 2-year-olds," *PloS one*, vol. 6, no. 4, p. e18746, 2011. 23

[117] X. Kan, H. Cui, J. Lukemire, Y. Guo, and C. Yang, "Fbnetgen: Task-aware gnn-based fmri analysis via functional brain network generation," in *International Conference on Medical Imaging with Deep Learning*. PMLR, 2022, pp. 618–637. 23, 71, 73

[118] T. D. Satterthwaite, M. A. Elliott, K. Ruparel, J. Loughead, K. Prabhakaran, M. E. Calkins, R. Hopson, C. Jackson, J. Keefe, M. Riley *et al.*, "Neuroimaging of the philadelphia neurodevelopmental cohort," *Neuroimage*, vol. 86, pp. 544–553, 2014. 23

[119] B. J. Casey, T. Cannonier, M. I. Conley, A. O. Cohen, D. M. Barch, M. M. Heitzeg, M. E. Soules, T. Teslovich, D. V. Dellarco, H. Garavan *et al.*, "The adolescent brain

cognitive development (abcd) study: imaging acquisition across 21 sites," *Developmental cognitive neuroscience*, vol. 32, pp. 43–54, 2018. 23

[120] J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar *et al.*, "Functional network organization of the human brain," *Neuron*, vol. 72, no. 4, pp. 665–678, 2011. 23

[121] M. F. Glasser, S. N. Sotiropoulos, J. A. Wilson, T. S. Coalson, B. Fischl, J. L. Andersson, J. Xu, S. Jbabdi, M. Webster, J. R. Polimeni *et al.*, "The minimal preprocessing pipelines for the human connectome project," *Neuroimage*, vol. 80, pp. 105–124, 2013. 23

[122] K. Dadi, M. Rahim, A. Abraham, D. Chyzhyk, M. Milham, B. Thirion, G. Varoquaux, A. D. N. Initiative *et al.*, "Benchmarking functional connectome-based predictive models for resting-state fmri," *NeuroImage*, vol. 192, pp. 115–134, 2019. 23, 72

[123] L. Liu, G. Wen, P. Cao, T. Hong, J. Yang, X. Zhang, and O. R. Zaiane, "Braintgl: A dynamic graph representation learning model for brain network analysis," *Computers in Biology and Medicine*, vol. 153, p. 106521, 2023. 23

[124] R. C. Craddock, G. A. James, P. E. Holtzheimer III, X. P. Hu, and H. S. Mayberg, "A whole brain fmri atlas generated via spatially constrained spectral clustering," *Human brain mapping*, vol. 33, no. 8, pp. 1914–1928, 2012. 23

[125] H. Cui, W. Dai, Y. Zhu, X. Kan, A. A. C. Gu, J. Lukemire, L. Zhan, L. He, Y. Guo, and C. Yang, "Braingb: a benchmark for brain network analysis with graph neural networks," *IEEE transactions on medical imaging*, vol. 42, no. 2, pp. 493–506, 2022. 23

[126] Y. Liu, L. He, B. Cao, P. Yu, A. Ragin, and A. Leow, "Multi-view multi-graph embedding for brain network clustering analysis," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018. 23

[127] L. Badea, M. Onu, T. Wu, A. Roceanu, and O. Bajenaru, "Exploring the reproducibility of functional connectivity alterations in parkinson's disease," *PLoS One*, vol. 12, no. 11, p. e0188196, 2017. 23, 26

[128] C. Yan and Y. Zang, "Dparsf: a matlab toolbox for" pipeline" data analysis of resting-state fmri," *Frontiers in systems neuroscience*, p. 13, 2010. 24

[129] P. S. Aisen, R. C. Petersen, M. C. Donohue, A. Gamst, R. Raman, R. G. Thomas, S. Walter, J. Q. Trojanowski, L. M. Shaw, L. A. Beckett *et al.*, "Clinical core of the alzheimer's disease neuroimaging initiative: progress and plans," *Alzheimer's & Dementia*, vol. 6, no. 3, pp. 239–246, 2010. 24

[130] P. S. Aisen, R. C. Petersen, M. Donohue, M. W. Weiner, A. D. N. Initiative *et al.*, "Alzheimer's disease neuroimaging initiative 2 clinical core: progress and plans," *Alzheimer's & Dementia*, vol. 11, no. 7, pp. 734–739, 2015. 24

[131] M. W. Weiner, D. P. Veitch, P. S. Aisen, L. A. Beckett, N. J. Cairns, R. C. Green, D. Harvey, C. R. Jack Jr, W. Jagust, J. C. Morris *et al.*, "The alzheimer's disease neuroimaging initiative 3: Continued innovation for clinical trial improvement," *Alzheimer's & Dementia*, vol. 13, no. 5, pp. 561–571, 2017. 24

[132] L. A. Beckett, M. C. Donohue, C. Wang, P. Aisen, D. J. Harvey, N. Saito, A. D. N. Initiative *et al.*, "The alzheimer's disease neuroimaging initiative phase 2: Increasing the length, breadth, and depth of our understanding," *Alzheimer's & Dementia*, vol. 11, no. 7, pp. 823–831, 2015. 25

[133] K. Marek, D. Jennings, S. Lasch, A. Siderowf, C. Tanner, T. Simuni, C. Coffey, K. Kieburtz, E. Flagg, S. Chowdhury *et al.*, "The parkinson progression marker initiative (ppmi)," *Progress in neurobiology*, vol. 95, no. 4, pp. 629–635, 2011. 25

[134] F. V. Farahani, W. Karwowski, and N. R. Lighthall, "Application of graph theory for identifying connectivity patterns in human brain networks: a systematic review," *frontiers in Neuroscience*, vol. 13, p. 585, 2019. 26

[135] S. Noble, D. Scheinost, E. S. Finn, X. Shen, X. Papademetris, S. C. McEwen, C. E. Bearden, J. Addington, B. Goodyear, K. S. Cadenhead *et al.*, "Multisite reliability of mr-based functional connectivity," *Neuroimage*, vol. 146, pp. 959–970, 2017. 28

[136] X. Li, P. S. Morgan, J. Ashburner, J. Smith, and C. Rorden, "The first step for neuroimaging data analysis: Dicom to nifti conversion," *Journal of neuroscience methods*, vol. 264, pp. 47–56, 2016. 28

[137] K. J. Gorgolewski, T. Auer, V. D. Calhoun, R. C. Craddock, S. Das, E. P. Duff, G. Flandin, S. S. Ghosh, T. Glatard, Y. O. Halchenko *et al.*, "The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016. 28

[138] M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich, and S. M. Smith, "Fsl," *NeuroImage*, vol. 62, no. 2, pp. 782–790, 2012, 20 YEARS OF fMRI. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811911010603 29

[139] R. W. Cox, "Afni: Software for analysis and visualization of functional magnetic resonance neuroimages," *Computers and Biomedical Research*, vol. 29, no. 3, pp. 162–173, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010480996900142 29

[140] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, "A reproducible evaluation of ants similarity metric performance in brain image registration," *Neuroimage*, vol. 54, no. 3, pp. 2033–2044, 2011. 29

[141] B. Fischl, "Freesurfer," *NeuroImage*, vol. 62, no. 2, pp. 774–781, 2012, 20 YEARS OF fMRI. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811912000389 29

[142] M. D. Fox, A. Z. Snyder, J. L. Vincent, M. Corbetta, D. C. Van Essen, and M. E. Raichle, "The human brain is intrinsically organized into dynamic, anticorrelated functional networks," *Proceedings of the National Academy of Sciences*, vol. 102, no. 27, pp. 9673–9678, 2005. 29

[143] M. D. Fox, D. Zhang, A. Z. Snyder, and M. E. Raichle, "The global signal and observed anticorrelated resting state brain networks," *Journal of neurophysiology*, vol. 101, no. 6, pp. 3270–3283, 2009. 29

[144] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1. Oakland, CA, USA, 1967, pp. 281–297. 30

[145] A. Abraham, F. Pedregosa, M. Eickenberg, P. Gervais, A. Mueller, J. Kossaifi, A. Gramfort, B. Thirion, and G. Varoquaux, "Machine learning for neuroimaging with scikit-learn," *Frontiers in neuroinformatics*, vol. 8, p. 14, 2014. 30, 83, 100

[146] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963. 30

[147] X. Zhang, J. Huang, Y. Yang, X. He, R. Liu, and N. Zhong, "Applying python in brain science education," in *2019 International Joint Conference on Information, Media and Engineering (IJCIME)*. IEEE, 2019, pp. 396–400. 31

[148] S. Parisot, S. I. Ktena, E. Ferrante, M. Lee, R. G. Moreno, B. Glocker, and D. Rueckert, "Spectral graph convolutions for population-based disease prediction," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2017, pp. 177–185. 31

[149] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998. 32

[150] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. 32

[151] T. Bonald, N. de Lara, Q. Lutz, and B. Charpentier, "Scikit-network: Graph analysis in python." *J. Mach. Learn. Res.*, vol. 21, no. 185, pp. 1–6, 2020. 32

[152] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016. 32, 43, 46, 62, 72, 94, 97

[153] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1025–1035. 32, 43, 46, 62, 72

[154] X. Bresson and T. Laurent, "Residual gated graph convnets," *arXiv preprint arXiv:1711.07553*, 2017. 32, 62

[155] V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson, "Benchmarking graph neural networks," *arXiv preprint arXiv:2003.00982*, 2020. 32, 122, 128

[156] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014. 33

[157] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in neural information processing systems*, vol. 2, 1989. 33

[158] J. Xu, J. Ni, and Y. Ke, "A class-aware representation refinement framework for graph classification," *Information Sciences*, vol. 679, p. 121061, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025524009757 38, 79

[159] X. Song, R. Ma, J. Li, M. Zhang, and D. P. Wipf, "Network in graph neural network," *arXiv preprint arXiv:2111.11638*, 2021. 39

[160] P. A. Papp, K. Martinkus, L. Faber, and R. Wattenhofer, "Dropgnn: random dropouts increase the expressiveness of graph neural networks," *Advances in Neural Information Processing Systems*, vol. 34, 2021. 39

[161] M. Ding, J. Tang, and J. Zhang, "Semi-supervised learning on graphs with generative adversarial nets," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 913–922. 39

[162] J. Byrd and Z. Lipton, "What is the effect of importance weighting in deep learning?" in *International Conference on Machine Learning*. PMLR, 2019, pp. 872–881. 39

[163] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988. 39

[164] J. Baek, M. Kang, and S. J. Hwang, "Accurate learning of graph representations with graph multiset pooling," *arXiv preprint arXiv:2102.11533*, 2021. 41

[165] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," *Advances in neural information processing systems*, vol. 30, 2017. 42

[166] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *computer vision and pattern recognition*, 2016. 42

[167] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, 1958. 43, 61, 78, 96

[168] D. Q. Nguyen, T. D. Nguyen, and D. Phung, "Universal graph transformer self-attention networks," *arXiv preprint arXiv:1909.11855*, 2019. 43, 46

[169] V. Vapnik, "The nature of statistical learning theory (information science and statistics) springer-verlag," *New York*, 2000. 44

[170] P. L. Bartlett and W. Maass, "Vapnik-chervonenkis dimension of neural nets," *The handbook of brain theory and neural networks*, pp. 1188–1192, 2003. 44, 115

[171] M. Kabkab, E. Hand, and R. Chellappa, "On the size of convolutional neural networks and generalization performance," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 3572–3577. 45, 115

[172] K. Kersting, N. M. Kriege, C. Morris, P. Mutzel, and M. Neumann, "Benchmark data sets for graph kernels," 2016. [Online]. Available: http://graphkernels.cs.tu-dortmund.de 46

[173] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *Advances in neural information processing systems*, vol. 33, pp. 22 118–22 133, 2020. 46

[174] A. Wijesinghe and Q. Wang, "A new perspective on" how graph neural networks go beyond weisfeiler-lehman?"," in *International Conference on Learning Representations*, 2021. 46

[175] A. Nouranizadeh, M. Matinkia, M. Rahmati, and R. Safabakhsh, "Maximum entropy weighted independent set pooling for graph neural networks," *arXiv preprint arXiv:2107.01410*, 2021. 47, 62

[176] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987. 52

[177] C. Thornton, "Separability is a learner's best friend," in *4th Neural Computation and Psychology Workshop, London, 9–11 April 1997*. Springer, 1998, pp. 40–46. 52

[178] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection-theory and algorithms," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 43. 52

[179] J. Xu, Q. Bian, X. Li, A. Zhang, Y. Ke, M. Qiao, W. Zhang, W. K. J. Sim, and B. Gulyás, "Contrastive graph pooling for explainable classification of brain networks," *IEEE Transactions on Medical Imaging*, pp. 1–1, 2024. 55, 71, 97

[180] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011. 62, 96

[181] S.-J. Weng, J. L. Wiggins, S. J. Peltier, M. Carrasco, S. Risi, C. Lord, and C. S. Monk, "Alterations of resting state functional connectivity in the default network in adolescents with autism spectrum disorders," *Brain research*, vol. 1313, pp. 202–214, 2010. 66

[182] M. Assaf, K. Jagannathan, V. D. Calhoun, L. Miller, M. C. Stevens, R. Sahl, J. G. O'Boyle, R. T. Schultz, and G. D. Pearlson, "Abnormal functional connectivity of default mode sub-networks in autism spectrum disorder patients," *Neuroimage*, vol. 53, no. 1, pp. 247–256, 2010. 66, 83

[183] R. K. Kana, E. B. Sartin, C. Stevens Jr, H. D. Deshpande, C. Klein, M. R. Klinger, and L. G. Klinger, "Neural networks underlying language and social cognition during self-other processing in autism spectrum disorders," *Neuropsychologia*, vol. 102, pp. 116–123, 2017. 66, 83

[184] R. L. Gould, R. G. Brown, A. M. Owen, E. T. Bullmore, and R. J. Howard, "Task-induced deactivations during successful paired associates learning: an effect of age but not alzheimer's disease," *Neuroimage*, vol. 31, no. 2, pp. 818–831, 2006. 66

[185] M. R. Brier, J. B. Thomas, A. Z. Snyder, T. L. Benzinger, D. Zhang, M. E. Raichle, D. M. Holtzman, J. C. Morris, and B. M. Ances, "Loss of intranetwork and internetwork resting state functional connections with alzheimer's disease progression," *Journal of Neuroscience*, vol. 32, no. 26, pp. 8890–8899, 2012. 66, 101

[186] P. Alexopoulos, C. Sorg, A. Förschler, T. Grimmer, M. Skokou, A. Wohlschläger, R. Perneczky, C. Zimmer, A. Kurz, and C. Preibisch, "Perfusion abnormalities in mild cognitive impairment and mild dementia in alzheimer's disease measured by pulsed arterial spin labeling mri," *European archives of psychiatry and clinical neuroscience*, vol. 262, pp. 69–77, 2012. 66

[187] S. Tu, S. Wong, J. R. Hodges, M. Irish, O. Piguet, and M. Hornberger, "Lost in spatial translation–a novel tool to objectively assess spatial disorientation in alzheimer's disease and frontotemporal dementia," *Cortex*, vol. 67, pp. 83–94, 2015. 66

[188] O. Monchi, M. Petrides, J. Doyon, R. B. Postuma, K. Worsley, and A. Dagher, "Neural bases of set-shifting deficits in parkinson's disease," *Journal of Neuroscience*, vol. 24, no. 3, pp. 702–710, 2004. 66

[189] N. J. Gerrits, Y. D. van der Werf, K. M. Verhoef, D. J. Veltman, H. J. Groenewegen, H. W. Berendse, and O. A. van den Heuvel, "Compensatory fronto-parietal hyperactivation during set-shifting in unmedicated patients with parkinson's disease," *Neuropsychologia*, vol. 68, pp. 107–116, 2015. 66

[190] M. A. Fernández-Seara, E. Mengual, M. Vidorreta, G. Castellanos, J. Irigoyen, E. Erro, and M. A. Pastor, "Resting state functional connectivity of the subthalamic nucleus in p arkinson's disease assessed using arterial spin-labeled perfusion f mri," *Human brain mapping*, vol. 36, no. 5, pp. 1937–1950, 2015. 66

[191] Y. H. Chan, W. C. Yew, and J. C. Rajapakse, "Semi-supervised learning with data harmonisation for biomarker discovery from resting state fmri," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2022, pp. 441–451. 72, 108

[192] J. Xu, A. Zhang, Q. Bian, V. P. Dwivedi, and Y. Ke, "Union subgraph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 14, 2024, pp. 16 173–16 183. 73

[193] P. Li, Y. Wang, H. Wang, and J. Leskovec, "Distance encoding: Design provably more powerful neural networks for graph representation learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4465–4478, 2020. 77, 91

[194] H. Wang, H. Yin, M. Zhang, and P. Li, "Equivariant and stable positional encoding for more powerful graph neural networks," *arXiv preprint arXiv:2203.00199*, 2022. 77, 91

[195] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *Proceedings of the Web Conference 2021*, 2021, pp. 2069–2080. 79

[196] R. Li, X. Wu, A. S. Fleisher, E. M. Reiman, K. Chen, and L. Yao, "Attention-related networks in alzheimer's disease: A resting functional mri study," *Human brain mapping*, vol. 33, no. 5, pp. 1076–1088, 2012. 83

[197] S. Rombouts, F. Barkhof, C. Van Meel, and P. Scheltens, "Alterations in brain activation during cholinergic enhancement with rivastigmine in alzheimer's disease," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 73, no. 6, pp. 665–671, 2002. 83

[198] S. Frisch, J. Dukart, B. Vogt, A. Horstmann, G. Becker, A. Villringer, H. Barthel, O. Sabri, K. Müller, and M. L. Schroeter, "Dissociating memory networks in early alzheimer's disease and frontotemporal lobar degeneration-a combined study of hypometabolism and atrophy," *PloS one*, vol. 8, no. 2, p. e55251, 2013. 83

[199] C. Ecker, A. Marquand, J. Mourão-Miranda, P. Johnston, E. M. Daly, M. J. Brammer, S. Maltezos, C. M. Murphy, D. Robertson, S. C. Williams *et al.*, "Describing the brain

in autism in five dimensions—magnetic resonance imaging-assisted diagnosis of autism spectrum disorder using a multiparameter classification approach," *Journal of Neuroscience*, vol. 30, no. 32, pp. 10 612–10 623, 2010. 83

[200] K. J. Worsley, C. H. Liao, J. Aston, V. Petre, G. Duncan, F. Morales, and A. C. Evans, "A general statistical analysis for fmri data," *Neuroimage*, vol. 15, no. 1, pp. 1–15, 2002. 88

[201] X. Wang, J. Chen, B. T. Dai, J. Xin, Y. Gu, and G. Yu, "Effective graph kernels for evolving functional brain networks," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 150–158. 89

[202] M. S. Burtsev, Y. Kuratov, A. Peganov, and G. V. Sapunov, "Memory transformer," *arXiv preprint arXiv:2006.11527*, 2020. 92

[203] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, "Vision transformers need registers," in *The Twelfth International Conference on Learning Representations*, 2023. 92

[204] W. Cheney and D. Kincaid, "Linear algebra: Theory and applications," *The Australian Mathematical Society*, vol. 110, pp. 544–550, 2009. 92

[205] B. T. Yeo, F. M. Krienen, J. Sepulcre, M. R. Sabuncu, D. Lashkari, M. Hollinshead, J. L. Roffman, J. W. Smoller, L. Zöllei, J. R. Polimeni *et al.*, "The organization of the human cerebral cortex estimated by intrinsic functional connectivity," *Journal of neurophysiology*, 2011. 100

[206] F. Agosta, M. Pievani, C. Geroldi, M. Copetti, G. B. Frisoni, and M. Filippi, "Resting state fmri in alzheimer's disease: beyond the default mode network," *Neurobiology of aging*, vol. 33, no. 8, pp. 1564–1578, 2012. 101

[207] J. S. Damoiseaux, K. E. Prater, B. L. Miller, and M. D. Greicius, "Functional connectivity tracks clinical deterioration in alzheimer's disease," *Neurobiology of aging*, vol. 33, no. 4, pp. 828–e19, 2012. 101

[208] M. D. Shen, P. Shih, B. Öttl, B. Keehn, K. M. Leyden, M. S. Gaffrey, and R.-A. Müller, "Atypical lexicosemantic function of extrastriate cortex in autism spectrum disorder: evidence from functional and effective connectivity," *Neuroimage*, vol. 62, no. 3, pp. 1780–1791, 2012. 101

[209] H. Koshino, R. K. Kana, T. A. Keller, V. L. Cherkassky, N. J. Minshew, and M. A. Just, "fmri investigation of working memory for faces in autism: visual coding and underconnectivity with frontal areas," *Cerebral cortex*, vol. 18, no. 2, pp. 289–300, 2008. 101

[210] A. Padmanabhan, C. J. Lynch, M. Schaer, and V. Menon, "The default mode network in autism," *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, vol. 2, no. 6, pp. 476–486, 2017. 101

[211] R. A. Cooper, F. R. Richter, P. M. Bays, K. C. Plaisted-Grant, S. Baron-Cohen, and J. S. Simons, "Reduced hippocampal functional connectivity during episodic memory retrieval in autism," *Cerebral Cortex*, vol. 27, no. 2, pp. 888–902, 2017. 102

[212] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of machine learning research*, vol. 17, no. 59, pp. 1–35, 2016. 108

[213] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 443–450. 108

[214] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017. 108

[215] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019. 108

[216] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization," *arXiv preprint arXiv:1911.08731*, 2019. 108

[217] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville, "Out-of-distribution generalization via risk extrapolation (rex)," in *International conference on machine learning*. PMLR, 2021, pp. 5815–5826. 108

[218] Y.-X. Wu, X. Wang, A. Zhang, X. He, and T.-S. Chua, "Discovering invariant rationales for graph neural networks," *arXiv preprint arXiv:2201.12872*, 2022. 108

[219] S. Miao, M. Liu, and P. Li, "Interpretable and generalizable graph learning via stochastic attention mechanism," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15 524–15 543. 108

[220] Y. Chen, Y. Zhang, Y. Bian, H. Yang, M. Kaili, B. Xie, T. Liu, B. Han, and J. Cheng, "Learning causally invariant representations for out-of-distribution generalization on graphs," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 131–22 148, 2022. 108

[221] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli, "Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 104–112. 113

[222] J. Cadena, A. K. Vullikanti, and C. C. Aggarwal, "On dense subgraphs in signed network streams," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 51–60. 113

[223] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data*. AMLBook New York, 2012, vol. 4. 115

[224] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 122, 129

[225] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017. 122, 129

[226] W. Tang, Q. Zhu, X. Gong, C. Zhu, Y. Wang, and S. Chen, "Cortico-striato-thalamo-cortical circuit abnormalities in obsessive-compulsive disorder: a voxel-based morphometric and fmri study of the whole brain," *Behavioural brain research*, vol. 313, pp. 17–22, 2016. 127

[227] R. Min, J. Cheng, T. Price, G. Wu, and D. Shen, "Maximum-margin based representation learning from multiple atlases for alzheimer's disease classification," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014: 17th International Conference, Boston, MA, USA, September 14-18, 2014, Proceedings, Part II 17*. Springer, 2014, pp. 212–219. 128

[228] R. Min, G. Wu, J. Cheng, Q. Wang, D. Shen, and A. D. N. Initiative, "Multi-atlas based representations for alzheimer's disease diagnosis," *Human brain mapping*, vol. 35, no. 10, pp. 5052–5070, 2014. 128

[229] M. Liu, D. Zhang, D. Shen, and A. D. N. Initiative, "View-centralized multi-atlas classification for alzheimer's disease diagnosis," *Human brain mapping*, vol. 36, no. 5, pp. 1847–1865, 2015. 128

[230] M. Liu, D. Zhang, and D. Shen, "Relationship induced multi-template learning for diagnosis of alzheimer's disease and mild cognitive impairment," *IEEE transactions on medical imaging*, vol. 35, no. 6, pp. 1463–1474, 2016. 128