University of Texas at Arlington

MavMatrix

Computer Science and Engineering Dissertations

Computer Science and Engineering Department

Spring 2024

When Brain Meets Artificial Intelligence

Lu Zhang
The University of Texas at Arlington

Follow this and additional works at: https://mavmatrix.uta.edu/cse_dissertations

Part of the Artificial Intelligence and Robotics Commons, Bioinformatics Commons, Computational Neuroscience Commons, Data Science Commons, and the Disease Modeling Commons

Recommended Citation

Zhang, Lu, "When Brain Meets Artificial Intelligence" (2024). *Computer Science and Engineering Dissertations*. 2.
https://mavmatrix.uta.edu/cse_dissertations/2

This Dissertation is brought to you for free and open access by the Computer Science and Engineering Department at MavMatrix. It has been accepted for inclusion in Computer Science and Engineering Dissertations by an authorized administrator of MavMatrix. For more information, please contact leah.mccurdy@uta.edu, erica.rousseau@uta.edu, vanessa.garrett@uta.edu.

WHEN BRAIN MEETS ARTIFICIAL INTELLIGENCE

by

LU ZHANG

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2024

Copyright © by LU ZHANG 2024

All Rights Reserved

To my dear family, for all their endless trust, continuous support, and unconditional love.

ACKNOWLEDGEMENTS

My six years at UTA have been an invaluable life experience. I began my Ph.D. journey in 2018 as a dreamer, fascinated by brain science and intelligence. With a blend of nerves and excitement, I embarked on a path of ideals, passion, the unknown, and challenges. As I near the end of this journey, I've reached a significant milestone. Looking back at yesterday from this point, I realize that I've encountered many invaluable mentors and friends along the way. They are the most precious treasures of my journey.

I would like to thank my advisor, Dr. Dajiang Zhu, for his unwavering support throughout my doctoral journey. Under his expert guidance, I developed essential skills in research methodology, paper writing, and presentation delivery, which have been instrumental in my growth as a researcher. The collaborative brainstorming sessions in Dr. Zhu's office, where the whiteboard served as our creative canvas, are some of my most cherished memories. One particularly memorable occasion occurred during my first year of doctoral studies, on the eve of the MICCAI deadline. Dr. Zhu generously devoted his time to assist me and another lab member, Akib, in refining our paper. Together, we worked tirelessly until 2:00 am, striving to achieve perfection in our work. To streamline communication and avoid unnecessary email exchanges, Dr. Zhu would simply call out our names from his office when discussion was needed. Situated next door, we were always ready to race to his office at lightning speed.

I would like to thank to Dr. Jean Gao, Dr. Junzhou Huang, and Dr. Li Wang for generously agreeing to serve on my Ph.D. committee. I am very fortunate to have had the opportunity to collaborate with the three esteemed professors over the past six years, resulting in the publication of multiple impactful papers. Their expertise, guidance, and invaluable insights have played a pivotal role in shaping my doctoral journey. I am deeply appreciative of their unwavering support in guiding my research and academic pursuits. Their contributions have not only enriched my work but have also been instrumental in fostering my growth as a scholar. I am truly grateful for their invaluable contributions to my academic success.

I am very thankful to receive support from Dr. Tianming Liu, Dr. Tuo Zhang, and Dr. Feng Liu. I had the privilege of meeting Dr. Tianming Liu during the 2019 MICCAI conference, and since then, we have maintained a close collaboration. As a distinguished professor and esteemed scholar, Dr. Liu has offered invaluable guidance in my research endeavors, for which I am deeply appreciative. Before joining the PhD program, I had the valuable opportunity to study neuroimaging preprocessing under the guidance of Dr. Tuo Zhang, which laid a solid foundation for my PhD studies. Dr. Feng Liu is my alumni, and I had the privilege of meeting him during the first year of my PhD study. Since then, we have nurtured a lasting friendship. His invaluable advice has been immensely helpful in guiding my career path.

I would like to extend my heartfelt gratitude to my friends Xiao Shi and Zeyu Zhang. Over the past six years, their companionship and support have given me the strength to overcome a lot of obstacles and challenges. I will never forget the moments we spent together in the office and at home, working on assignments, discussing problems, watching movies, and cooking meals. The memory of Zeyu parading around in his torn sweatpants with holes in the backside, and Xiao and I eventually deciding to throw them away, will always bring a smile to my face. There are countless cherished memories that I will hold dear forever. I wish my dear friends all the best in the days ahead.

I am sincerely grateful to all my lab members: Xiaowei, Yanjun, Chao Cao, Jing Zhang, Tom, Haotian, Junqi, and Akib. We have stayed up late together before deadlines, collaborated on projects, and celebrated New Year's with lively parties. Without the support of my colleagues in our lab, I wouldn't have accomplished what I have. Their creativity and unwavering support have truly enriched my doctoral journey, making it an unforgettable experience.

Due to space limit, I regret that I cannot list all the names of those who have helped and supported me throughout my Ph.D. studies. However, I want to express my heartfelt gratitude to all of them.

Finally, I wish to thank my parents, Shijun Zhang and Congling Dong, for their endless love and unwavering support which have been a constant source of power throughout my life and academic journey. I am forever grateful for their sacrifices, guidance, and encouragement that have shaped me into the person I am today. I would also like to thank my boy friend Dong Wang, for his love and unwavering support during my Ph.D. study.

April, 26, 2024

ABSTRACT

WHEN BRAIN MEETS ARTIFICIAL INTELLIGENCE

LU ZHANG, Ph.D.

The University of Texas at Arlington, 2024

Supervising Professor: Dr. Dajiang Zhu

When we review the history of development of artificial intelligence (AI), we will find that brain science plays a pivotal role in fostering breakthroughs in AI, such as artificial neural networks (ANNs). Today, AI has made remarkable strides, particularly with the emergence of large language models (LLMs), surpassing expectations and achieving human-level performance in certain tasks. Nonetheless, an insurmountable gap remains between AI and human intelligence. It is urgent to establish a bridge between brain science and AI, promoting their mutual enhancement and collaborations. This involve establishing connections from brain science to AI (brain-inspired AI), and reversely, from AI to brain (AI in brain science).

For this purpose, in this thesis, we aim to 1) take advantages of the powerful capabilities of cutting-edge AI techniques to explore the intricacies of the brain, including both normal brain and brain disorders; and 2) leverage the superior organizational principles of brain networks to inspire and guide the design of AI models. My research in the two directions has opened new frontiers for brain science and AI research: 1)Exploring the foundational organizational principles of human brain with AI techniques. We develop expressive and effective deep learning models that

can capture the non-trivial brain structure–function relationship at individual level; 2) Identifying imaging-based biomarkers of brain dementia and modeling the continuous brain disease progression using AI techniques. We develop multi-modal deep neural networks to integrate multiple types of brain network connectome and characterize their deep relationship as an "individual connectome signature" for brain disease study. We also propose novel structure learning methods to model the continuum of disease progression; 3) LLMs in neuroscience and healthcare domain. LLMs are at the forefront of AI, which have change the paradign of model design. To explore how to unlock the potential benefits of LLMs in neuroscience and improve healthcare outcomes, we have developed effective approaches to tailor these models to the unique requirements of the specialized domains. 4) Pioneering research in brain-inspired AI. We conduct post hoc analysis to explore the connections between artificial neural networks (ANNs) and biological neural networks (BNNs), laying a solid foundation for brain-inspired AI. Building upon these findings, we proactively instill the organizational principles observed in BNNs into ANNs and develop more powerful ANNs.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

ABSTRACT vii

LIST OF ILLUSTRATIONS xv

LIST OF TABLES xxxi

Chapter Page

1. INTRODUCTION 1
    1.1 Motivation 1
    1.2 Thesis Outline 4
2. RECOVERING BRAIN STRUCTURAL CONNECTIVITY FROM FUNCTIONAL CONNECTIVITY VIA MULTI-GCN BASED GENERATIVE ADVERSARIAL NETWORK 7
    2.1 Introduction 7
    2.2 Materials and Methods 12
        2.2.1 Participants and Data Description 12
        2.2.2 Data Preprocessing 13
        2.2.3 Generation of Functional Connectivity and Structural Connectivity 13
        2.2.4 Method Overview 14
        2.2.5 Graph Convolutional Network (GCN) 15
        2.2.6 Multi-GCN Based GAN (MGCN-GAN) 19
        2.2.7 Structure-Preserving (SP) Loss Function 20
    2.3 Results 22

2.3.1 Experimental Setting 22
2.3.2 Predicted Structural Connectivity 23
2.3.3 Evaluation the predicted SC using different atlases 30
2.3.4 Model Evaluation 30
2.3.5 Comparison with other widely used methods 35
2.4 Discussion 36
2.4.1 Outliers in the experiments 36
2.4.2 Limitations and Future work 37
2.5 Conclusion 39
3. DEEP FUSION OF BRAIN STRUCTURE-FUNCTION IN MILD COGNITIVE IMPAIRMENT 40
3.1 Introduction 41
3.2 Materials and Methods 46
3.2.1 Methods Overview 46
3.2.2 Functional Profile Learning 46
3.2.3 Brain Structure-Function Fusion 48
3.2.4 Brain Network Convolution 51
3.2.5 MCI-NC Classification 52
3.3 Results 54
3.3.1 Participants 54
3.3.2 Data Description and Preprocessing 55
3.3.3 Experimental Setting 55
3.3.4 Classification Performance 56
3.3.5 Deep Brain Connectome 57
3.3.6 Model Evaluation 64
3.4 Discussion 67

3.5 Conclusion 69

4. DISEASE2VEC: ENCODING ALZHEIMER'S PROGRESSION VIA DISEASE EMBEDDING TREE 70

4.1 Introduction 71

4.2 Materials and Methods 75

4.2.1 Data 75

4.2.2 Method Overview 76

4.2.3 Disease Embedding Learning 76

4.2.4 Ordered Embedding Constraint 79

4.3 Results 81

4.3.1 Experimental Setting 81

4.3.2 Classification Performance 82

4.3.3 The Learned Disease Embedding Tree 85

4.3.4 Ablation Study 89

4.3.5 Reproducibility and Generalizability 93

4.4 Discussion and Conclusion 95

4.4.1 Advantages 96

4.4.2 Limitations and Future Work 98

5. CORTEX2VEC: ANATOMICAL EMBEDDING OF CORTICAL FOLDING 100

5.1 Introduction 101

5.2 Materials and Methods 104

5.2.1 Datasets Description and Data Pre-processing 104

5.2.2 Identification of GyralNet and 3HGs 104

5.2.3 Multi-Hop Features Encoding 107

5.2.4 Learning-based Embedding Framework 108

5.2.5 Evaluation of the Embedding Effectiveness 110

5.3 Results 111
5.3.1 Experimental Setting 112
5.3.2 Effectiveness of ROI Embeddings 112
5.3.3 Effectiveness of 3HGs Individual Embeddings 115
5.3.4 Regression Performance 119
5.4 Discussion 120
5.5 Conclusion 123
6. LARGE LANGUAGE MODELS (LLMS) IN HEALTHCARE DOMIAN . 124
6.1 Evaluation of LLMs in Clinical NLP 124
6.1.1 Datasets 125
6.1.2 Methods 126
6.1.3 Results 141
6.1.4 Discussion 142
6.1.5 Conclusion 145
6.2 Exploring the Trade-Offs: Unified Large Language Models vs Local Fine-Tuned Models for Highly-Specific Radiology NLI Task 146
6.2.1 Datasets 146
6.2.2 Method 149
6.2.3 Experiment and Analysis 155
6.2.4 Discussion and Conclusion 161
6.3 DeID-GPT: Zero-shot Medical Text De-Identification by GPT-4 . . . 163
6.3.1 Datasets 164
6.3.2 Methods 165
6.3.3 Experiment & Results 172
6.3.4 Discussion and Conclusion 176
6.3.5 Future Perspective 179

7. A UNIFIED AND BIOLOGICALLY-PLAUSIBLE RELATIONAL GRAPH REPRESENTATION OF VISION TRANSFORMERS 181
7.1 Introduction 182
7.2 Relational Graph in Vision Transformers 184
    7.2.1 Structure of Vision Transformer Models 184
    7.2.2 Aggregation and Affine Graphs 185
    7.2.3 Construction of Aggregation Relational Graph in Vision Transformers 187
    7.2.4 Aggregation Graph between Layers 189
7.3 Experiments 189
    7.3.1 Downstream Task of Pretrained ViT Models 190
    7.3.2 ViT Model Training from Scratch 191
    7.3.3 Graph Similarities between ViTs and BNNs 193
7.4 Result 193
    7.4.1 Relationship between Aggregation Graph Measures and Classification Performances 193
    7.4.2 Dynamic Changing Characteristics of Graph Measures during Model Training 194
    7.4.3 Graph Measure Similarity between ViTs and BNNs 195
    7.4.4 Aggregation Graph on Model Design 195
7.5 Discussion 198
    7.5.1 Fine-tune Partial vs. Full Model 198
    7.5.2 Graph Structures with Different Sampling Efficiencies 198
    7.5.3 Aggregation Graphs between Layers 200
    7.5.4 Impact on Model Design and Optimization of ANNs 200
    7.5.5 Limitation 201

7.6 Conclusion 202

8. CORE-PERIPHERY PRINCIPLE GUIDED REDESIGN OF SELF-ATTENTION IN TRANSFORMERS 203

8.1 Introduction 203

8.2 Methods 207

8.2.1 Related Work 207

8.2.2 Core-Periphery Principle Guided Transformer 209

8.3 Results 215

8.3.1 Exploring Core-Periphery Graphs 215

8.3.2 Sweet Spots for CP-ViTs 218

8.3.3 Visualization of Important Patches 223

8.3.4 Fast Search for Sweet Spots 224

8.4 Conclusion 227

9. DISCUSSION AND CONCLUSION 228

9.1 Conclusion 228

9.2 Future Work 230

REFERENCES 231

BIOGRAPHICAL STATEMENT 272

LIST OF ILLUSTRATIONS

Figure Page

2.1 An illustration of the proposed multi-GCN based generative adversarial network (MGCN-GAN). Firstly, by using brain atlas (Destrieux Atlas [1] and Desikan-Killiany Atlas [2] along with diffusion MRI and rs-fMRI data, we extracted the averaged BOLD signal of each brain region. Then we constructed functional connectivity (FC) by different methods (correlation, partial correlation, threshold FC, and binarized FC) and structural connectivity (SC) by diffusion MRI derived fiber counts. SC was used as 1) ground truth to guide the generator at the beginning of the training process; 2) real samples of discriminator. FC was used as: 1) features associated with the nodes; 2) initialization of the GCN topology. The features and topology were fed into generator to predict SC. The predicted SC were used to 1) iteratively update the GCN topology and 2) train discriminator as fake samples. The whole model is trained based on the proposed structure preserving (SP) loss function 11

2.2 Illustration of the graph convolution process. A graph $\mathcal{G}$ can be represented by an adjacency matrix $\mathcal{A}$ and a feature matrix $\boldsymbol{X}$. The GCN takes the two matrices as input to conduct graph convolution. We used the red node as an example to show the convolution process. The neighbors with the same order have the same color in graph $\mathcal{G}$. The colors of features are the same as the corresponding nodes. For the lth layer, the red edges of the input adjacency matrix $\mathcal{A}$ indicate the neighbors that participate in the convolution process and the features of these activate neighbors are non-transparent 18

2.3 (a1-a2): Comparison of the predicted SCs and real SCs of 20 randomly selected subjects in HCP (a1) and ADNI (a2) datasets. For both datasets, we showed the 10 real SC matrices (the first row) and the corresponding 10 predicted SC matrices (the second row). Each column belongs to the same subject. Two patches of the matrices are extracted from the same location and their enlarged patches are showed in the middle. (b1-b2): Comparison of the predicted SCs and real individual SCs after subtracting the populaion-averaged SC. To better visualize the individual variability, the population-averaged SC was subtracting for each of the forty matrices in (a1) and (a2). The brain connectivity was generated via Destrieux atlas 25

2.4 We quantitatively measured the similarity between predicted SCs with real SCs (based on Destrieux atlas) from three ways (magnitude, overall pattern and network property) by six measures (Mean Square Error (MSE), cosine similarity, Pearson correlation coefficient (PCC), mean degree, mean strength and mean clustering coefficient) of HCP (a) and ADNI (b) datasets. For each measure, two sets of values are calculated: one is the deviation of the predictions from the real SC and the other is the deviation of the averaged SC from the real SC. We showed the two sets of values by line chart and displayed the distributions by violin plot. The significance analysis was also conducted with p-value calculated by one tail two sample T-test 27

2.5 (a1-a2): Comparison of the predicted SCs and real SCs of the same 20 subjects as used in Fig. 2.3. The brain connectivity was generated via Desikan-Killiany atlas. Each column belongs to the same subject. For each subject, we showed the real SC matrix in the first row and the predicted SC matrix in the second row. Two patches of the matrices are extracted from the same location and their enlarged patches are showed in the middle. (b1-b2): Comparison of the predicted SCs and real individual SCs after subtracting the population-averaged SC. To better visualize the individual variability, the population-averaged SC was subtracting for each of the forty matrices in (a1) and (a2) 28

2.6 We quantitatively measured the similarity between the predicted SCs with real SCs (based on Desikan-Killiany atlas) from three ways (magnitude, overall pattern and network property) by six measures (Mean Square Error (MSE), cosine similarity, Pearson correlation coefficient (PCC), mean degree, mean strength and mean clustering coefficient) of HCP (a) and ADNI (b) datasets. For each measure, two sets of values are calculated, one is the deviation of the predictions from the real SC and the other is the deviation of the averaged SC from the real SC. We showed the two sets of values by line chart and displayed the distributions by violin plot. The significance analysis was also conducted with p-value calculated by one tail two sample T-test 29

2.7 Results of different generator architectures for HCP dataset (a) and ADNI dataset (b). $\mathrm{GCN}\,(\mathrm{G}_1\|\mathrm{G}_2\|\ldots\|\mathrm{G}_\mathrm{k})$ represents the architecture of generator. The generator is composed of k two-layer GCNs, and the output feature dimension of the first layer of ith GCN is G_i 32

2.8 Results of different initializations of the learnable coefficients from HCP dataset (a) and ADNI dataset (b). $\boldsymbol{\theta_i}$ is the initialization of the learnable coefficients and $\boldsymbol{\theta_c}$ is the corresponding converged value 34

2.9 Results of MGCN-GAN with different loss functions on HCP dataset (a) and ADNI dataset (b) 35

3.1 Illustration of the proposed Deep Brian Connectome learning based on Graph Based Deep Model (GBDM). By using Destrieux atlas [1] along with DTI and resting state fMRI data, we extracted the averaged BOLD signal of each brain region (148 regions in total) (a) and constructed brain structural network (b). Then the structural network was used as the initialization of graph topology, and the functional information was used by GBDM to iteratively update the topology of the graph – deep brain connectome (c), to maximize its classification power for MCI patients 45

3.2 The major steps of the proposed Graph Based Deep Model (GBDM). The model consists of four components: 1) functional profile learning, 2) brain structure-function fusion, 3) brain network convolution and 4) MCI-NC classification. Functional profile learning aims to learn a new representation of pairwise functional relations by introducing a trainable functional mapping matrix. In the step of brain structure-function fusion, structural network and the learned functional profile are combined as a hybrid brain network and it will be updated in each iteration. Using functional connectivity (defined with Pearson's correlation coefficient) as features associated to the network nodes (brain regions), we conduct graph convolution based on the hybrid brain network. A fully connected network is appended at last for MCI-NC classification and the results are used to update the parameters in previous layers. Because the learned hybrid brain network "deeply" integrates brain structure-function information to maximize its MCI-NC classification power, we name it as Deep Brain Connectome 50

3.3 Multi-layer graph convolutional network. According to the input adjacency matrix $\hat{A}$ and feature matrix P, a GCN layer creates a hidden representation for each node by combining features from its neighbor nodes based on W_l. After the combination, a nonlinear transformation is applied to the hidden representation. By stacking multiple layers, the final hidden representation of each node gathers information from both direct neighbors and high-order neighbors (indirectly connected via other nodes) 53

3.4 The influences of using different numbers of GCN layers for aggregating features. We use step to represent the number of edges in a shortest path connecting two nodes in the graph. The nodes with the same color represent that they have the same steps from the center node (red). With more GCN layers are stacked the deeper of brain network relations are considered 53

3.5 The results of the learned Deep Brain Connectome and comparison with brain structural network. (a) shows the averaged results for NC group (the first row) and MCI group (the second row). For each group, the first column displays the group-wise structural network and the order of the brain regions listed in the matrix is the same as Destrieux atlas. The second column shows the learned deep brain connectome. Compared to brain structural network, the increased and decreased connectivity in deep brain connectome are shown in the third and fourth column. The top changed connectivity using the thresholds of 70% (for increased connectivity) and 37% (for decreased connectivity) are shown in the fifth and sixth column. (b) shows the increased and decreased connectivity of 6 randomly selected subjects from NC group (the first block) and MCI group (the second block) 59

3.6 Changing rate (%) for the increased connectivity (a) and decreased connectivity (b) of NC and MCI groups 61

3.7 The top increased/decreased connectivity and the involved brain regions for both MCI and NC groups. (a) shows the brain regions related to the top 10 increased and decreased connectivity and their connections. (b) shows the brain regions related to the increased and decreased connectivity with an averaged changing rate above 72% and 38%, respectively. For subfigures in both (a) and (b), the first row shows the involved regions and the second row highlights the regions as colored bubbles as well as the connections 63

3.8 Top increased and decreased connectivity under different model settings. We used different combinations of σ and feature dimension to conduct our experiments 66

3.9 Influence of σ and feature dimension on changing scale of the altered connectivity. (a) shows the maximum and median of changing rate of the increased and decreased connectivity with a spectrum of σ (from 1.0 to 3.0) based on $\mathrm{G}(148, 296)$. (b) shows the maximum and median of changing scale of the increased and decreased connectivity of different feature dimensions with the same $\sigma = 2.0$ 66

4.1 Training: we used functional connectivity as input and learned a Disease Embedding Tree (DETree) to model the entire progression of AD in the latent space. In the tree structure, each small bubble represents a single subject, and the colors indicate different clinical groups, including normal control - NC (green), significant memory concern - SMC (yellow), early MCI - EMCI (orange), late MCI - LMCI (pink) and AD (red). Each edge in the DETree indicates that the connecting two nodes have higher similarity in the latent space. The five larger bubbles represent the learned group embeddings. Prediction: During the prediction, new patients will be projected into the latent space which are represented as scattered bubbles. The color of the bubble indicates the true label, the location of the bubble shows its state in the entire development process from NC to AD, and the prediction of the bubble is based on the nearest group embedding 73

4.2 (a): Visualization of the feature distribution of different clinical groups in the embedding space. (b): DETree learned from multiple clinical groups, including NC, SMC, EMCI, LMCI, and AD. Each small bubble in the tree represents a single subject color-coded according to their clinical group. Each edge indicates higher similarity between connected nodes. The five larger bubbles represent the group level embeddings. (c): the Mini-Mental State Examination (MMSE) score [3] mapped to DETree. (d): the Alzheimer's Disease Assessment Scale – Cognitive subscale (ADAS-cog) score [4] mapped to DETree. The small bubbles in (c) and (d) correspond to the small bubbles in (b) at the same location 86

4.3 Top connectivity that contributes most to the learned DETree structure. In each block, the top and bottom rows display the involved brain regions and connectivity, respectively 88

4.4 (A) and (B): Classification performance of various dimensions of the embedding space in two testing dataset settings. (C): The differences in F_1 and Acc measures of two settings 90

4.5 Visualization of the feature distribution of different clinical groups in the embedding space with varying dimensions: NC - green, SMC - yellow, EMCI - orange, LMCI - pink, and AD - red 91

4.6 Different DETree structures learned in the embedding space with varying dimensions. The blue circles are used to highlight the embeddings from different clinical groups with small distance 92

4.7 Classification performance of various dimensions of the embedding space based on dataset2 94

4.8 Visualization of the feature distribution of different clinical groups in the embedding space with varying dimensions based on dataset2 . . . 94

4.9 Different DETree structures learned in the embedding space with varying dimensions based on dataset2 95

5.1 Schematic of the proposed embedding framework. **(a)** Identification of GyralNet and 3HGs. **a1** White matter cortical surface mapped by gyral altitude; **a2** Segmentation of gyral crest (white regions) from sulci basins (color regions); **a3** Connecting the gyral crest regions into a completed graph by tree marching (black curves). A magnification view of the circled patch is displayed between a2 and a3; **a4** Pruning the redundant branches to preserve the main trunk of the graph (black curves) - GyralNet; **a5** Identification of 3HGs (labeled by green bubbles). **(b)** 3HG's multi-hop feature encoding. **b1** Parcellating the entire cortex into 75 ROIs via Destrieux Atlas and assigning each 3HG with an ROI label as node feature; **b2** Numerically representing each ROI label by one-hot encoding; **b3** By considering multi-hop neighbors, 3HG are encoded by multi-hop features. **(c)** The proposed learning-based embedding framework (details can be found in Section 5.2.4) 105

5.2 Evaluation of the learned ROI embeddings using strength of 3HG's multi-hop connections (defined in Section 5.2.5). (a) The strength of 3HG's multi-hop connections calculated by the whole population with 1064 subjects; (b) The cosine similarity of each pair of the ROI embedding vectors learned by different multi-hop features (1-hop features, 2-hop features, and 3-hop features, respectively). For the matrices in both (a) and (b), the order of the brain regions is the same as the order defined in Destrieux atlas [1], where most of the first 44 regions are gyri while the last 31 regions are sulci 113

5.3 Cross-subject correspondences of 3HGs 117

5.4 Cross-subject individuality and variability. We used three 3HGs (three blocks: top, middle, and bottom) as examples to show their correspondences in different subjects. We randomly selected 12 subjects and adopted the cosine similarity of 0.9 as the threshold. The corresponding 3HGs are represented by bubbles and the cosine similarity was encoded by the color of the bubbles. Due to the individuality, it is possible to find zero, one or multiple correspondences across different subjects . . 118
6.1 Zero shot prompt and one shot prompt used in the experiment. 127
6.2 Five shot prompt used in the experiment. 128
6.3 Major world-wide LLMs and their relationships 140
6.4 Prompts used in the experiments. Modifications for IRSA and CoT are highlighted 150
6.5 We assessed the impact of various prompt methods (as detailed in Sec. 6.2.3.1) on the performance of ChatGPT and GPT-4 for both zero-shot and few-shot scenarios. 156
6.6 We compared the performance of ChatGPT, GPT-4, and five other fine-tuned LLMs. To evaluate the impact of prompt examples on ChatGPT and GPT-4, we tested their performance with zero-shot and 10-shot examples, where greater example usage resulted in higher accuracy. Additionally, we explored the effect of increasing the size of the training dataset for the fine-tuned LLMs, which led to improved accuracy. . . . 159
6.7 Sample original clinical note from the 2014 i2b2/UTHealth dataset. . . 165
6.8 Sample de-identified clinical note from the 2014 i2b2/UTHealth dataset. 166
6.9 The left column shows the main components of the ChatGPT API code, while the right column shows the steps involved in generating an anonymized example. 167

6.10 To anonymize clinical notes using the ChatGPT API, we first supply prompts from the system and then send the clinical notes from the user role. The returned results from ChatGPT will be anonymized by the model, and detected sensitive PHI information will be replaced by the term 'redacted'. 168

6.11 This is the optimal prompt template we have designed for this task. We break down a prompt into 3 main segments: The task statement, the command, and specific rules. The task statement describes the goal and scope of the task. The command specifies actions to complete the task. Specific rules define specific requirements and optionally contain concrete examples. In the figure, we also annotate each line with a tag that describes its identity or correspondence in the PHI categories. . . 170

6.12 We present 4 examples of bad prompts. Example1: Only stating the task in the prompt is not optimal; Example2: Punctuation matters. The extra period made the command uninterpreted by ChatGPT; Example3: Multiple-task statement confuses ChatGPT; Example4: Did not explicitly specify the desired output 172

7.1 The aggregation and affine graph in different ViT models. (a) The patch embedding in vision transformer models. (b) The aggregation graph and (d) the affine graph in vision transformer models. (c) the structure of transformer vision models 185

7.2 Relationship between two aggregation graph measures and classification performance of 21 ViT models on four datasets, Larger spot size represents larger model parameters. 190

7.3 Dynamic change of the two graph measures during ViT-Ti model training from scratch on ImageNet-1k dataset. 192

7.4 Similarity of relational graphs between vision models and BNNs. L: the average path length, C: the clustering coefficient. 192

7.5 The accuracy and graph loss value during training when the ImageNet-1k pretrained ViT-T was as target model and the ImageNet-21k pretrained ViT-B was as reference model. 196

7.6 The accuracy changed on ImageNet-1k after ViT models trained with graph loss. (left) The reference model also pretrained on ImageNet-1k. (right) The reference model pretrained on ImageNet-21k. 197

7.7 The prediction accuracy on Flower17 with fine-tuning partial and full models 198

7.8 The graph measures of different training methods in the same model . 199

7.9 The aggregation graphs between layers in different ViT models. (a) The clustering coefficients of Mixer, MetaFormer, ViT and Swin. (b) The visualization of aggregation graphs in ViT-B. 199

8.1 The Core-Periphery principle in brain networks inspires the design of ANNs. The Core-Periphery structure broadly exists in brain networks, with a dense “core” of nodes (pink) densely interconnected with each other and a sparse “periphery” of nodes (blue) sparsely connected to the core and among each other. Inspired by this principle of BNN, we aim to instill the Core-Periphery structure into the self-attention mechanism and propose a new CP-ViT model 206

8.2 Core-Periphery Principle Guided Re-design of Self-Attention. The proposed Core-Periphery guided re-design framework for ViTs consists of two major components: the Core-Periphery graph generator and the re-design of the self-attention mechanism. The basic idea is that we mapped the ViT structure to graphs and proposed a new graph representation paradigm to represent the self-attention mechanism. Under this paradigm, the design of the self-attention mechanism can be turned into a task of designing desirable graphs. (a) The CP graph generator was proposed to generate graphs with Core-Periphery property in a wide range of search spaces. (b) The self-attention of the nodes is controlled by the generated CP graph and the patches are re-distributed to different nodes by a novel patch distribution method. (c) The new self-attention mechanism will upgrade the regular self-attention in vanilla ViT. The new ViT architecture is thus named as CP-ViT 209

8.3 Illustration of Patch Redistribution Process. The pink nodes are the core nodes, while the blue nodes are the periphery nodes. The initial patch distribution at the first epoch is the same as the vanilla ViTs. After each iteration during the training process, the gradients of patches discriminate from each other due to different contributions to the classification. The red the image patches are, the high gradient they are. Thus, the core patches that contribute most to the classification task are re-distributed to core nodes 213

8.4 (a) Two types of representative brain networks in motor and working memory tasks. (b) Three examples of CP graphs. (c) Complete graph. The first row in (a), (b), and (c) shows their wiring patterns, while the second row shows their corresponding adjacency matrices. Black color in adjacency matrices means connections between nodes, while white represents no edge. (d) Graph search space defined by the total nodes number and the core nodes number. The complete graphs are located at the diagonal highlighted by a red box and the CP graphs are located at the remaining parts 215

8.5 Performance of CP-ViT measured using INbreast, CIFAR-10, CIFAR-100 and TinyImageNet datasets. Sub-figures on the left column under each datasets show the top 1 classification accuracy of the CP-ViTs and vanilla ViTs in the search space. A deeper color means higher top 1 accuracy. Sweet spots are marked by red crosses, in which CP-ViTs achieve better performance than vanilla ViT. Sub-figures on the middle column are the accuracy degradation of the CP-ViTs compared to vanilla ViTs. Sub-figures on the right column are the self-attention connection ratio of the CP-ViTs and vanilla ViT. Lighter color means a lower connection ratio. Sweet spots are marked by the blue crosses . 220

8.6 Visualization of important image patches that were distributed to the core nodes. For the INbreast dataset (the first block), images of three randomly selected subjects for each class were shown. For each subject, there are three images displayed in three columns. The left column is the original image, the middle column shows the important patches marked by red, and the right column is the eye gaze of medical physicians on the image. For the natural image datasets (the second block, CIFAR-10, CIFAR-100 and TinyImageNet), the important patches identified in eight randomly selected classes were displayed. The left column is the original image, and the right column shows the identified core patches marked in red 225

8.7 Visualization of Core-periphery measures versus the classification performance. The regression results of the normalized independent probability versus the classification accuracy for experiments on each dataset are presented in (a), (b), (c), and (d). The core-periphery measures for brain networks of motor and working memory are shown in (e) and (f) 226

LIST OF TABLES

Table Page

2.1 Comparison with other widely used methods. 36

3.1 Performance Comparison of Different Methods in AD/MCI Classification. 58

3.2 Classification Performance under Different Model Settings. 65

4.1 Classification Performance of DETree and Four Traditional Machine Learning Methods. The orange shade and blue shade highlight the best and the second-best results, respectively. 83

4.2 Classification Performance of DETree and Other Deep Learning Methods. cMCI/pMCI: MCI patient who converted to AD within 36 months; sMCI: MCI patients who didn't convert to AD within 36 months. EMCI/LMCI: early/late MCI. The orange shade and blue shade highlight the best and the second-best results, respectively. 84

5.1 Similarity between ground truth matrix and embedding similarity matrix. 115

5.2 Regression performance of 2-hop embedding. 120

6.1 Test Results for Compared LLMs 141

6.2 Sample Size of RadQNLI Dataset 148

6.3 Modalities and Organs Covered by RadQNLI Dataset 149

6.4 The accuracy(ACC) of the five recent LLMs tested on proposed RadQNLI dataset with zero-shot incontext learning. Red and blue denotes the best and the second-best results, respectively. 159

6.5 We adopted ROUGE scores to set different levels of inference difficulty and assessed ChatGPT's performance on tasks of varying levels of difficulty. 161
6.6 Monetary costs ($) and inference times (s/item) of ChatGPT and GPT4 on RadQNLI testset. 162
6.7 List of HIPAA identifiers. 168
6.8 Clinical notes de-identification results with LLMs. 175
8.1 Evaluation of the Core-Periphery property in CP graphs, graphs generated by other graph generators, and brain networks 218
8.2 Summary of datasets 218
8.3 Comparison between the proposed CP-ViT in sweet spots with finetuned vanilla ViT-S [5]. * means vanilla ViT-S finetuned by ourselves. 221
8.4 Comparisons with state-of-the-art transformers and other architectures. 223

CHAPTER 1

INTRODUCTION

1.1 Motivation

When we review the history of development of artificial intelligence (AI), we will find a close connection between brain science and AI. The earliest substantial work in the field of AI was done in the mid-20th century by the British logician and computer pioneer Alan Mathison Turing. In one of his papers [6], Turing asked “can machines think”? He aims to create machines with human-level or even higher intelligence, which are capable of performing a wide range of intellectual tasks that humans can do. This is recognized as the beginning of AI. Therefore, AI has a close connection with brain from its very outset. Later, the microstructure inside the human brain observed by the light microscopy inspired the perceptron [7], which is the foundation of multi-layer artificial neural networks (ANNs). The convolution property and multi-layer structure of brain visual cortex inspired the convolutional neural networks (CNN), the dominant architecture in the computer vision (CV) domain before the emergence of transformer architecture [8, 9]. The attention mechanism in the human brain, discovered using the positron emission tomography (PET) imaging, inspired the attention module [10], which is a key component in transformer. The working memory discovered using functional magnetic resonance imaging (fMRI) inspired the memory module in deep learning models, contributing to the development of long short-term memory (LSTM) [11]. Although there have also been some explorations of AI that are independent of brain science, the latter plays a pivotal role in fostering breakthroughs during the development of AI. The research findings in brain science reveal important insights related to the principles of intelligence, which can inspire significant theoretical and technological advancements in AI community.

On the other hand, the relationship between brain science and AI is not unidirectional; it's not just about brain science inspiring AI. AI techniques can also serve as powerful tools to advance the comprehension of the brain. The human brain is widely considered as one of the most intricate and advanced information-processing systems in the world, with billions of neurons connecting into an exceptionally complex network that allows for the proliferation of intelligence. Conventional computational methods, such as graph theory and statistical modeling, are limited in their ability to process highly complex brain networks. AI techniques, including machine learning, deep learning, and the latest large language models (LLMs), have show significant superiority in capturing underlying patterns within complex input data, achieving notable successes across various domains. Particularly noteworthy are LLMs, the latest generation of AI, which have change the paradigm of model design. LLMs are primarily pretrained on general-domain datasets, and their performance in specialized fields, such as brain science or neuroscience, remains incompletely evaluated. The application of LLMs in these domains faces new challenges. Therefore, exploring how to effectively apply advanced AI techniques, including both traditional ANNs and LLMs, to the field of brain science to better study and comprehend the brain is a crucial research topic.

Another important problem is how to leverage the prior knowledge and organizational principles discovered in brain to inspire the development of new ANNs and boost their performance. As our understanding of the brain advances, we will discover new superior organizational principles within brain networks. How to effectively integrate these principles into ANNs remains an open question. Currently, there have been some attempts for this problem. For example, You et al. [12] conducted qualitatively post-hoc analysis and found that the best-performing ANNs surprisingly resemble biological neural networks (BNNs), which indicates that ANNs and BNNs may share some common principles to achieve optimal performance in either machine learning tasks or cognition/behavior processes. Although this work provided new insights about connections between BNNs and ANNs, it didn't explore how to instill the organization principles of BNNs into ANNs.

To address these challenges and bridge the brain science and AI to foster their mutual enhancement and collaboration, we conducted a series research work, focusing on three major problems: 1) AI in brain science. In this part, we tailored/developed novel AI models to explore the foundational organizational principles of human brain and investigate various brain disorders. 2) LLMs in neuroscience and healthcare. In this part, we conducted a series research work to explore how to unlock the potential benefits of LLMs in neuroscience and improve healthcare outcomes. We have developed effective approaches to tailor these LLMs to the unique requirements of specialized domains. 3) Brian-inspired-AI. In this part, we firstly conducted post hoc study to discuss the commonalities shared by BNNs and ANNs. Then based on the findings, we further proactively instilled the general organizational principles of brain networks into ANNs and designed new ANN architectures.

In general, brain science can offer valuable insights into the development of AI from the perspective of biological intelligence. Conversely, AI can sever as a powerful computational tool to deepen our understanding of brain, including both normal brain and brain disorders. This thesis paves the road for future studies in bridging the two domains.

1.2 Thesis Outline

The thesis focuses on three major problems: 1) AI in brain science, discussed in **Chapter 2**, **Chapter 3**, and**Chapter 4** ; 2) LLMs in neuroscience and healthcare, explored in **Chapter 5** and **Chapter 6**; and 3) Brian-inspired-AI, investigated in **Chapter 7** and **Chapter 8**. The details of each chapters are as follows:

In **Chapter 2**, we introduced the application of ANNs into exploring the fundamental organizational principles of brain networks. We integrated multiple graph convolutional networks (GCNs) into generative adversarial network (GAN) as graph filters and proposed a novel graph-based GAN model – MGCN-GAN. MGCN-GAN was used to model the relationship between brain structural network and functional network at individual level. Our findings indicate that there may exist a common regulation between specific brain structural and functional architectures across individuals.

In **Chapter 3**, we investigated the ANNs in the study of the brain disease – Alzheimer's Disease (AD). Jointly analyzing multimodal brain imaging data provides an irreplaceable opportunity to leverage the complementary information across different modalities for an in-depth study of brain disorders. In this chapter, we introduced a novel graph-based deep neural network – Deep Connectome, to simultaneously model individual structural and functional networks for brain disease analysis. The major advantage of the Deep Connectome is its capability to integrate multiple types of network connectome and characterize their deep relationship as an "individual connectome signature".

Another significant challenge in the study of brain diseases, particularly neurodegenerative disorders such as Alzheimer's disease (AD), arises from the extended preclinical phase, which spans from clinical asymptomatic stages to severe impairment. Current classification models that enforce a binary decision over two or multiple clinical groups may be suboptimal due to overlooking the continuous nature of the disease progression. To address this problem, in **Chapter 4**, we developed a novel structure learning method to model the continuum of AD progression as a tree-based trajectory. With the learned tree-based representation of AD progression, we can accurately predict a new individual patient's clinical status by projecting its brain networks back to this tree structure and examining the projected location within the entire AD spectrum. Moreover, the proposed framework is a general one that can be applied to various diseases.

In **Chapter 5**, we delved into the tokenization and embedding of brain networks. Tokenization and Embedding serve as the foundation for LLMs. They provide a way to divide the input data into basic units and represent them in a unified representation form that the model can understand and learn from. However, different from regular text or image data, the elements within brain networks, including the network nodes and edges, are typically specialized landmarks or locations on the cerebral cortex, posing challenges for direct tokenization and embedding. In this chapter, we leveraged the intrinsic relationships between the nodes within brain networks to introduce a novel indirect embedding method. Utilizing this method, we successfully addressed a longstanding correspondence problem in neuroscience.

In **Chapter 6**, we explored the application of LLMs in healthcare domain. LLMs are primarily pretrained on general-domain datasets, and their performance in specialized fields such as healthcare remains incompletely evaluated. To address this problem and to unlock the potential benefits of LLMs in improving healthcare outcomes, in this chapter, we conducted extensive assessments across various real-world tasks in healthcare domain. Through these evaluations, we scrutinized a key question: whether model designs should be generic or domain specific.

In **Chapter 7**, we adopted post hoc analysis to compare the similarity between the graph structures of Vision transformers (ViTs) and various real BNNs. Specifically, we proposed a novel unified and biologically plausible relational graph representation of ViTs. We explored the relevance between the relational graph structures of ViTs with their predictive performance and found that the model performance is closely related to the graph structures. We also compared the relational graph structures of ViTs with those of real BNNs derived from brain science and found that relational graph representations of top-performing ANNs have high similarity with real BNNs. In addition, our experimental results indicate that models similar with the more advanced biological brain networks perform better.

The findings in **Chapter 7** inspired us to proactively instill organization principles of BNNs into ANNs. In **Chapter 8**, we leveraged the Core-Periphery (CP) organization, which is widely existed in human brain networks and plays an important role in the efficienct information transmission, to guide the information communication mechanism in the self-attention of ViT and propose a novel CP-ViT model. In CP-ViT, the attention operation between nodes is defined by a sparse graph with a Core-Periphery organization, where the core nodes are redesigned and reorganized to play an integrative role and serve as a center for other periphery nodes to exchange information.

In **Chapter 9**, we concluded the whole dissertation and discussed future works.

CHAPTER 2

RECOVERING BRAIN STRUCTURAL CONNECTIVITY FROM FUNCTIONAL CONNECTIVITY VIA MULTI-GCN BASED GENERATIVE ADVERSARIAL NETWORK

We started by discussing the application of ANNs to explore the fundamental structure-function relationship in brain networks. Uncovering the non-trivial brain structure–function relationship is fundamentally important for revealing organizational principles of human brain. However, it is challenging to infer a reliable relationship between individual brain structure and function, e.g., the relations between individual brain structural connectivity (SC) and functional connectivity (FC). Brain structure–function displays a distributed and heterogeneous pattern, that is, diverse functionalities arise from non-overlapping sets of anatomical connections. This complex relation can be interwoven with widely existed individual structural and functional variations. Motivated by the advances of generative adversarial network (GAN) and graph convolutional network (GCN) in the deep learning field, in this chapter, we proposed a multi-GCN based GAN (MGCN-GAN) to infer individual SC based on corresponding FC by automatically learning the complex associations between individual brain structural and functional networks.

2.1 Introduction

A fundamental question in neuroscience is how to understand structure-function relationship of human brain. It is widely believed that brain structural architecture provides the substrate where rich functionality arises from, and therefore, the dynamics of brain function are closely related to the relatively fixed structure organization. Numerous studies confirmed that brain structure can determine, at least partially, brain functional patterns. For example, the concept of “connectional fingerprint” [13] suggests that each brain’s cytoarchitectonic area has a unique set of extrinsic inputs and outputs, which largely determine the function that each brain area performs. This close relationship between structural connection pattern and brain function has been confirmed and replicated in many literatures. For example, our previous work [14, 15, 16] proved that the same functional regions tend to possess consistent structural connectivity patterns across different individuals and populations. Koch et al. [17] directly compared brain structural connectivity (SC) and functional connectivity (FC) and found that regions which directly linked by structural connectivity show high functional connectivity. Skudlarski et al. [18] reported a significant overall agreement between SC and FC by calculating the partial correlation between the two global matrices. Some other studies implemented computational models to study the brain structure-function relationship at macroscale [19, 20], mesoscale [21], and microscale [22]. A consistent result achieved by these studies is that strong functional interactions tend to be accompanied with strong structural connections. On the other hand, some studies also found that parts of the FC may be not supported by the underlying SC. Greicius et al. [23] studied the relations between SC and four default mode network (DMN) related brain regions and found that strong FC can still exist without direct SC. This may be due to several factors. Firstly, the complex indirect interactions may widely exist among different brain regions. The functional connections observed between regions with little or no direct structural connections may be mediated by indirect structural connections. Secondly, brain’s structure-function behaves under a distributed and heterogeneous pattern: at network level, many functional relationships arise from non-overlapping sets of anatomical connections [24], which means functional networks does not necessarily correspond to the underlying structural substrate with a simple node-to-node mapping. Therefore, how to represent and analyze the relationship between brain structural and functional network, especially at individual level, is still challenging.

The existing approaches that have been used to explore brain structure-function relationship can be broadly divided into two categories: the first approach is to conduct association analysis using correlation coefficient, which mainly focuses on simple and linear relationship between SC and FC [17, 18]. The second is to apply graph theory to both brain structural and functional network for quantitative analysis, such as small world property [25, 26, 27], modular structure property [28, 29], and rich-club organization [30, 31]. All these approaches have fundamentally advanced our understanding of the relationship between brain structure and function at population level, but they are limited in charactering individual variability in subject-specific brain network. In addition to the above two widely used strategies, some other computational models have also been developed to bridge the gap between structural network topology and the related function by examining their relations at multiple scale and resolution [19], modeling dynamics [22] and constructing local mm-scale networks using animal model [21]. However, because of brain's distributed and heterogeneous structure-function pattern, traditional methods are limited to represent the complex relationship between individual SC and FC.

Recent advances in deep learning have revolutionized the fields of machine learning [32, 33] and brought breakthroughs for computational neuroimaging field including reconstruction [34], segmentation [35], detection [36], and computer-aided diagnosis [37]. Among numerous deep learning models, graph convolutional network (GCN) [38] generalizes the convolutional operations from grid data to graph data and witnesses great success in brain network domain recently [39, 40, 41]. More importantly, the generative adversarial network (GAN) [42] provides an efficient way to revisit the complex relationship between brain structure and function: as a generative model, GAN can powerfully handle the brain's distributed and heterogeneous structure-function pattern. Moreover, compared to other generative models, GAN effectively converts the regression problem to a classification problem through the adversarial training scheme. In this way, an explicit regression loss function is unnecessary, and the criterion used to evaluate the performance of the predictions is implicitly learned from the data. This can be especially suitable for areas with insufficient prior knowledge, such as brain network.

In this work, we proposed a multi-GCN based generative adversarial network (MGCN-GAN)(Fig. 2.1) to learn individual SC from the corresponding individual FC. We adopted GAN to handle brain's distributed and heterogeneous pattern. To overcome the inherent unstable behavior of GAN [42] caused by the adversarial training scheme, we proposed a novel structure-preserving (SP) loss function to guide the generator to learn the intrinsic SC patterns more effectively. In order to capture the complex relationship buried in both direct and indirect brain connections, we constructed the generator and discriminator using GCN. However, traditional GCN has two limitations: First, comparing to widely used convolutional neural network (CNN) that has multiple filters to capture multiple feature spaces, conventional GCN only has one filter (weight matrix) in each GCN layer and therefore can only learn a single feature map. Second, the performance of GCN may gradually decrease with increasing number of layers [43] and which limits the power of learning by deepening the network as CNN does. To address these limitations, we designed a multi-GCN based generator that used multiple GCNs instead of one deep GCN to simultaneously capture underlying complex interactions in brain network and avoid the performance decay by stacking more layers in a single GCN. We tested our methods on two datasets: Human Connectome Project (HCP) dataset [31] and Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset [44]. Our results show that the proposed MGCN-GAN can generate reliable individual SC based on corresponding individual FC. More importantly, our results imply that there may exist a common regulation between specific brain structural and functional architectures across individuals.

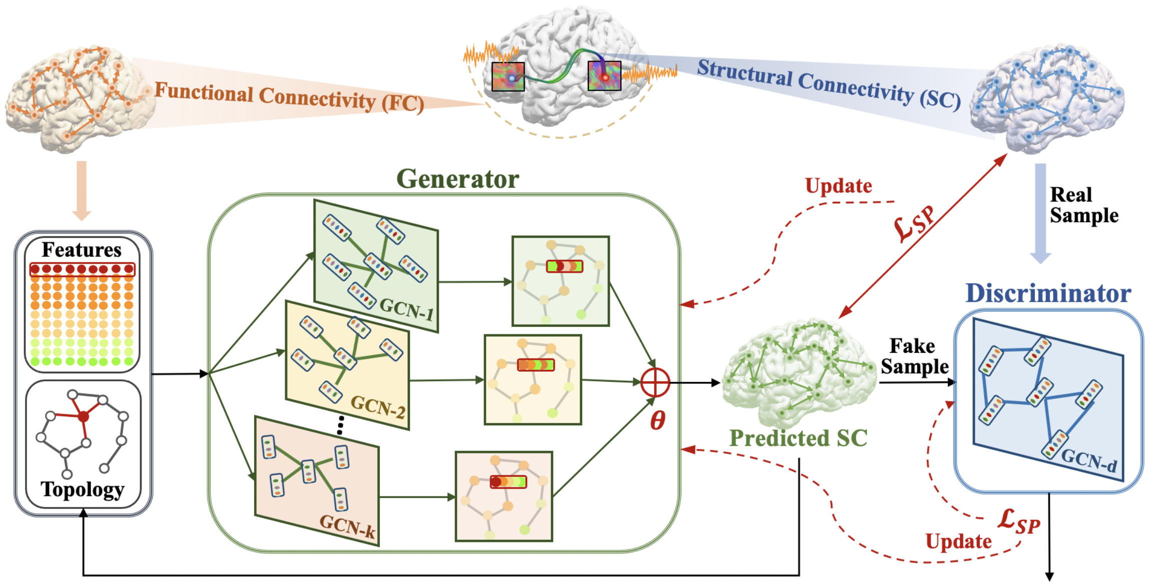


Figure 2.1. An illustration of the proposed multi-GCN based generative adversarial network (MGCN-GAN). Firstly, by using brain atlas (Destrieux Atlas [1] and Desikan-Killiany Atlas [2] along with diffusion MRI and rs-fMRI data, we extracted the averaged BOLD signal of each brain region. Then we constructed functional connectivity (FC) by different methods (correlation, partial correlation, threshold FC, and binarized FC) and structural connectivity (SC) by diffusion MRI derived fiber counts. SC was used as 1) ground truth to guide the generator at the beginning of the training process; 2) real samples of discriminator. FC was used as: 1) features associated with the nodes; 2) initialization of the GCN topology. The features and topology were fed into generator to predict SC. The predicted SC were used to 1) iteratively update the GCN topology and 2) train discriminator as fake samples. The whole model is trained based on the proposed structure preserving (SP) loss function.

Our proposed MGCN-GAN advances the state of the art in two ways: firstly, our model is designed to capture individual-specific structure-function relationship. Previous publication [45] found that similar structural damage of patients with the same pathology show different dysfunctions, which indicates the variability of individual structure–function relationship. Unveiling individual structure–function relationship is fundamentally important to the comprehensive understanding of individual variation in brain structure and function and is the premise and key step for personalized medicine. Secondly, we introduced multi-GCN architecture into GAN framework and designed a structure preserving (SP) loss function that help the model to generate high-quality SC. The MGCN-GAN is a flexible architecture with adjustable GCN components to fit different tasks with varying complexity.

2.2 Materials and Methods

2.2.1 Participants and Data Description

HCP dataset. In this work, we selected all the 1064 subjects which have structure MRI (T1-weighted), resting state fMRI (rs-fMRI) and diffusion MRI data from HCP S1200 release. For the T1-weighted MRI data, the Field of View (FOV) is 224 mm × 224 mm, voxel size is 0.7 mm isotropic, TR = 2.4 s, TE = 2.14 ms and flip angle = 8°. For the rs-fMRI data, the FOV is 208 mm × 180 mm, 72 slices, voxel size is 2.0 mm isotropic, TR = 0.72 s, TE = 33.1 ms, flip angle = 52° and there are 1200 volumes for each subject. For the diffusion MRI data, the gradient direction is 288 , the FOV is 210 mm × 180 mm, 111 slices, voxel size is 1.25 mm isotropic, TR = 5.52 s, TE = 89.5 ms and flip angle = 78°.

ADNI dataset. We used 132 normal control (CN) subjects (68 females, 64 males; 76.45 ± 7.68 yrs.) and 118 mild cognitive impairment (MCI) subjects (63 females, 55 males; 74.05 ± 8.29 yrs.) from ADNI dataset. Each subject has structure MRI (T1-weighted), rs-fMRI and diffusion MRI data. The FOV of T1-weighted MRI is 240 mm×256 mm×208 mm and the voxel size is 1.0 mm isotropic, TR = 2.3 s. The diffusion MRI data has 54 gradient directions, the FOV is 232 mm×232 mm×160 mm and the voxel size is 2.0 mm isotropic, TR = 7.2 s and TE = 56 ms. The rs-fMRI data has 197 volumes, the FOV is 220 mm × 220 mm × 163 mm, voxel size is 3.3 mm isotropic, TR = 3 s, TE = 30 ms and flip angle = 90°.

2.2.2 Data Preprocessing

We applied the same standard preprocessing procedures as in [46] to both HCP and ADNI datasets. In brief, we applied skull removal for all three modalities and registered T1 and fMRI to DTI space by FLIRT in FMRIB Software Library (FSL) [47]. For rs-fMRI images, we applied spatial smoothing, slice time correction, temporal pre-whitening, global drift removal and band pass filtering (0.01-0.1 Hz) via FEAT command in FSL. For DTI images, we applied eddy current correction using FSL and fiber tracking via MedINRIA [48]. For T1 images, we conducted segmentation using FreeSurfer package [49] and then adopted the Destrieux Atlas [1] and Desikan-Killiany Atlas [2] for ROI labeling.

2.2.3 Generation of Functional Connectivity and Structural Connectivity

For each subject, the whole brain is divided into 148/68 (148 for Destrieux Atlas and 68 for Desikan-Killiany Atlas) ROIs and represented as a network with 148/68 nodes. Averaged fMRI signal was calculated for each brain region and normalized by the standard Z-score normalization [50] formulated as:

$$\boldsymbol{f}_{\boldsymbol{i}} = \frac{\boldsymbol{f}_{\boldsymbol{i}} - \boldsymbol{f}_{\mu}}{\boldsymbol{f}_{\sigma}} (2.1)$$

where $\boldsymbol{f_i}$ is the averaged fMRI signal of brain region i, $\boldsymbol{f}_\mu$ and $\boldsymbol{f}_\sigma$ are the mean and the standard deviation of all 148 averaged fMRI signals. There exist several measurements to represent pairwise relationship between two fMRI derived BOLD signals, such as correlation [46], partial correlation [51] and covariance [52]. Since Pearson correlation coefficient (PCC) between the BOLD time series of two regions of interest derived from resting state fMRI data is the most used functional measurement to estimate functional connectivity [45], in this work we adopted PCC to construct functional connectivity (FC, denoted as $\boldsymbol{F} = [F_{i,j}] \in \boldsymbol{R}^{N\times N}$), where $F_{i,j} = \frac{\mathrm{cov}(\boldsymbol{f_i}, \boldsymbol{f_j})}{\sigma_{\boldsymbol{f_i}}\sigma_{\boldsymbol{f_j}}}$.

The structural connectivity (SC) was created in terms of fiber counts, denoted as $\boldsymbol{S} \in \boldsymbol{R}^{N\times N}$. $S_{i,j} \in R$ is the number of fibers connecting brain regions i and j. Then, we conducted normalization of S using (2.2) and (2.3).

$$S_{i,j} = \log_2 (S_{i,j} + 1)\,(2.2)$$

$$\boldsymbol{S} = \frac{\boldsymbol{S} - S_\mu}{S_\sigma}(2.3)$$

S_μ and S_σ are the mean and the standard deviation of $\boldsymbol{S}$. Because the fiber count can be a value from zero to a few thousands, which often follows a skewed distribution. Log transformation can equalize the standard deviations and make the distribution of the sample mean more consistent with a normal distribution [53]. Therefore, we first used log transformation to shrink the range of the fiber counts by (2.2) and then used (2.3) for normalization.

2.2.4 Method Overview

We proposed a Multi-GCN based GAN (MGCN-GAN) model to generate individual SC from the corresponding FC. Similar to vanilla GAN [42], MGCN-GAN is built on two components, i.e., generator and discriminator. To capture the highly complex relationship between SC and FC at the connectome level, we used multi-layer GCN architecture to design the generator and discriminator, namely Multi-GCN based generator and single-GCN based discriminator, respectively. Given an individual SC and the associated FC, the generator is trained to create real-like individual SC by competing with the discriminator based on an adversarial training scheme. The specific training steps are shown as follows: (i) FC is used as initial topology of brain network as well as features associated with the nodes; (ii) based on current topology, different GCN components of generator map the FC to different feature spaces to explore the latent relationship between SC and FC, and each GCN component outputs one feature matrix; (iii) all the output feature matrices are combined by learnable coefficients to generate the predicted SC in current iteration; (iv) discriminator acts as a classifier to differentiate the input SC as real SC (real samples) from the predicted SC (fake samples) generated by the generator; (v) the topology of the generator is updated by the predicted SC in the next iteration. Given the training data consisting of FC samples and their corresponding real SC samples, the whole model is trained based on the proposed SP loss function.

2.2.5 Graph Convolutional Network (GCN)

In many applications, data are generated from non-Euclidean domains and represented as graphs with complex interdependency and relationships between graph nodes. The complexity of graph data has imposed significant challenges on the existing deep learning algorithms, such as CNN model. Graph convolutional network (GCN) [38] extends traditional CNN by applying convolutional operations on graph-based instead of Euclidean-based neighbors and is essential to various applications. In this work, to represent the latent interactions between brain SC and FC, we adopt a multi-layer GCN architecture to build the proposed MGCN-GAN model. For the ease of better understanding GCN architecture, we first introduce the notations of a graph and the graph spectral convolution.

Graph. Let $\boldsymbol{\mathcal{G}} = (\boldsymbol{V}, \boldsymbol{\mathcal{E}})$ to be an undirected graph, where $\boldsymbol{\mathcal{V}} = \{v_1, v_2, \cdots, v_n\}$ is a set of vertices and $\mathcal{E} = \{e_{i,j} \mid i, j = 1, 2, \cdots, n\}$ is a set of edges. If there exists an edge connecting two vertexes v_i and v_j, then $e_{i,j} > 0$, otherwise, $e_{i,j} = 0$. Each vertex v_i can have its own attributes (features) which can be represented by a vector $\boldsymbol{x_i} \in \boldsymbol{R}^{1\times d}, d$ is the dimension of the attributes (features). $\boldsymbol{X} = [\boldsymbol{x}_1; \boldsymbol{x}_2; \cdots; \boldsymbol{x}_n] \in \boldsymbol{R}^{n\times d}$ is the feature matrix of graph $\boldsymbol{\mathcal{G}}$. The topology of $\mathcal{G}$ can be represented by an weighted adjacency matrix $\mathcal{A} = [a_{i,j}] \in \boldsymbol{R}^{n\times n}$, for all $i, j, a_{i,j} = e_{i,j}$. Thus, $\boldsymbol{G}$ also can be represented by $\mathcal{G} = (\mathcal{A}, \boldsymbol{X})$.

Graph SpectralConvolution Convolution. As shown in [54], the traditional convolution operators can be generalized to the graph setting by defining filters in the graph spectral domain. For a graph $\mathcal{G} = (\mathcal{A}, \boldsymbol{X})$ with the adjacency matrix $\mathcal{A} = [a_{i,j}] \in \boldsymbol{R}^{n\times n}$ and node-wise feature matrix $\boldsymbol{X} = [x_1; x_2; \cdots; x_n] \in \boldsymbol{R}^{n\times d}$, its normalized graph Laplacian is defined as $\boldsymbol{L} = \boldsymbol{I}_N - \mathcal{D}^{-\frac{1}{2}}\mathcal{A}\mathcal{D}^{-\frac{1}{2}}$, where $\boldsymbol{I_N}$ is the identity matrix and $\mathcal{D} = \sum_j a_{i,j}$ is the diagonal degree matrix. $\boldsymbol{L}$ can be decomposed as $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where $\boldsymbol{U}$ is the matrix of eigenvectors and $\boldsymbol{\Lambda}$ the diagonal matrix of eigenvalues. Then, the graph Fourier transform of a signal $\boldsymbol{x_i}$ can be expressed as $\boldsymbol{U}^T\boldsymbol{x_i}$. This allows to define the spectral convolution on graphs in the Fourier domain as $g_\theta * \boldsymbol{x_i} = \boldsymbol{U} g_\theta \boldsymbol{U}^T \boldsymbol{x_i}$. The spatial filter $g_\theta = \mathrm{diag}(\theta)$ can be regarded as a function of the eigenvalues of $\boldsymbol{L}$, i.e. $g_\theta(\boldsymbol{\Lambda})$, parameterized by a group of Fourier coefficients $\boldsymbol{\theta} \in R^N$. Calculating $\boldsymbol{U} g_\theta \boldsymbol{U}^T \boldsymbol{x_i}$ requires expensive computations $(O(N^2))$ and the eigendecomposition of $\boldsymbol{L}$ is prohibitively costly for large graphs. To tackle this problem, [55] used a polynomial in terms of Chebyshev expansion up to K-order to approximate the spectral convolution: $g_\theta(\boldsymbol{\Lambda}) \approx \sum_{k=0}^{K} \theta_k T_k(\widetilde{\boldsymbol{\Lambda}})$ with a rescaled $\widetilde{\boldsymbol{\Lambda}} = \frac{2}{\lambda_{\max}}\boldsymbol{\Lambda} - \boldsymbol{I}_N$ and $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{L}$. The Chebyshev polynomial is recursively defined as

$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, with $T_0 = 1$ and $T_1 = x$. Going back to the definition of the convolution of signal $\boldsymbol{x_i}$ with the filter g_θ, we now have: $g_\theta * \boldsymbol{x_i} \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{\boldsymbol{L}})\boldsymbol{x_i}$, with $\tilde{\boldsymbol{L}} = \frac{2}{\lambda_{\max}}\boldsymbol{L} - \boldsymbol{I}_N$. Since the K-order polynomials are exactly K-localized kernels which only takes nodes within K hops away from the central node into account for the convolution, thus, the complexity of calculating $\sum_{k=0}^{K} \theta_k T_k(\tilde{L})x_i$ is linear with the number of the edges. [38] further approximated $\lambda_{\max} = 2$ and gave the linear formulation of a GCN by $g_\theta * \boldsymbol{x_i} \approx \boldsymbol{\theta}\left(\boldsymbol{I}_N + \mathcal{D}^{-\frac{1}{2}}\mathcal{A}\mathcal{D}^{-\frac{1}{2}}\right)\boldsymbol{x_i}$. To this end, the graph spectral convolution can be carried out by a convolutional network with convolutional layers of the polynomial form. For example, a two-layer GCN was formulated as $Z = f(\mathcal{A}, \boldsymbol{X}) = f\left(\widehat{\mathcal{A}}\,\mathrm{ReLU}\left(\widehat{\mathcal{A}}XW^{(0)}\right)W^{(1)}\right)$ in [38], where $\widehat{\mathcal{A}} = \mathcal{D}^{-\frac{1}{2}}\mathcal{A}\mathcal{D}^{-\frac{1}{2}}$ is the Laplacian transformation of $\mathcal{A}$. In previous work [56], we compared $\widehat{\mathcal{A}} = \mathcal{A}$ with other three different Laplacian transformations of $\mathcal{A}$: 1) $\widehat{\mathcal{A}} = \mathcal{D} - \mathcal{A}$, 2) $\widehat{\mathcal{A}} = \mathcal{D}^{-\frac{1}{2}}\mathcal{A}\mathcal{D}^{-\frac{1}{2}}$, and 3) $\widehat{\mathcal{A}} = \mathcal{D}^{-1}\mathcal{A}$ and found that $\widehat{\mathcal{A}} = \mathcal{A}$ and $\widehat{\mathcal{A}} = \mathcal{D}^{-\frac{1}{2}}\mathcal{A}\mathcal{D}^{-\frac{1}{2}}$ give similar performances. Thus, in this work we directly used the functional connectivity to initialize the adjacency matrix ($\widehat{\mathcal{A}} = \boldsymbol{F}$) without Laplacian transformation, the reasons are as follows: 1) compared to $\widehat{\mathcal{A}} = \mathcal{D}^{-\frac{1}{2}}\boldsymbol{F}\mathcal{D}^{-\frac{1}{2}}, \widehat{\mathcal{A}} = \boldsymbol{F}$ needs less computational cost; and 2) in this work, we attempt to recover structural connectivity from functional connectivity, thus, the original functional connectivity matrix without extra transformation is a better choice.

Based on above discussion, the convolutional process of multi-layer graph convolutional network can be formulated as (2.4) and (2.5):

$$G(\mathcal{A}, \boldsymbol{X}, \boldsymbol{W}) = \sigma\left(\mathcal{A}\boldsymbol{H}^{l-1}\boldsymbol{W}^{l}\right) \quad (2.4)$$

$$\boldsymbol{H}^{l} = \begin{cases} \sigma\left(\mathcal{A}\boldsymbol{H}^{l-1}W^{l}\right), & l > 0 \\ \boldsymbol{X}, & l = 0 \end{cases} \quad (2.5)$$

where σ is the nonlinear activation function, $\boldsymbol{H}^l$ is the output of the lth convolution layer, $\boldsymbol{W}^l \in \boldsymbol{R}^{F_i \times F_o}$ is the weight matrix, F_i is the input feature size and F_0 is the output feature size. As shown in Fig. 2.2, W^l acts like a filter which selects related features from neighbors and defines how to combine these features. By stacking multiple graph convolutional layers, information from high-order neighbors (indirectly connected via other nodes) can be propagated along graph topology defined by the adjacency matrix $\mathcal{A}$. In this work, we represented brain as a graph, and took the individual FC as the feature matrix i. e., $\boldsymbol{X} = \boldsymbol{F}$ and the initialized topology $\boldsymbol{A_0} = \boldsymbol{F}$. By conducting graph-based convolution via the proposed MGCN-GAN model, we iteratively updated the graph topology and learned the individual SC.

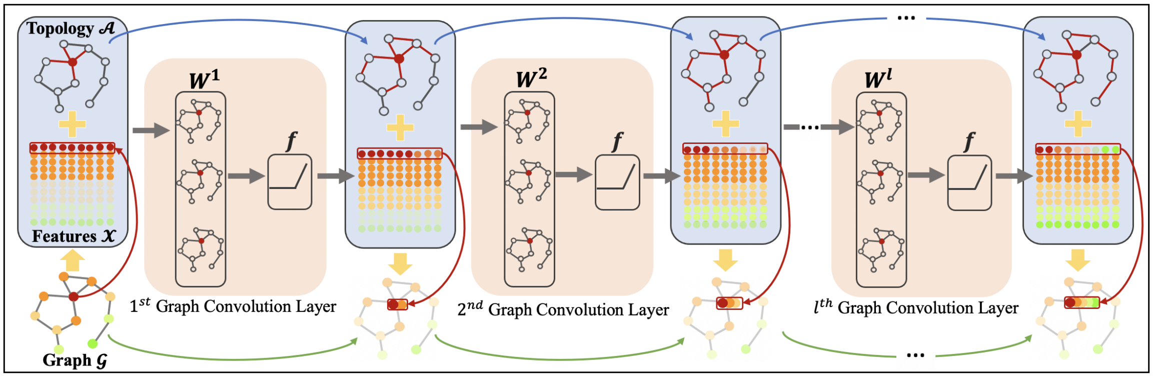


Figure 2.2. Illustration of the graph convolution process. A graph $\mathcal{G}$ can be represented by an adjacency matrix $\mathcal{A}$ and a feature matrix $\boldsymbol{X}$. The GCN takes the two matrices as input to conduct graph convolution. We used the red node as an example to show the convolution process. The neighbors with the same order have the same color in graph $\mathcal{G}$. The colors of features are the same as the corresponding nodes. For the lth layer, the red edges of the input adjacency matrix $\mathcal{A}$ indicate the neighbors that participate in the convolution process and the features of these activate neighbors are non-transparent.

2.2.6 Multi-GCN Based GAN (MGCN-GAN)

Multi-GCN based Generator. Inspired by the great success of CNN that uses multiple filters to identify different features, the proposed generator consists of multiple multi-layer GCNs. Different GCN components are designed for different feature spaces and each of them will learn a latent mapping from individual FC to its corresponding SC. Through paralleling multiple GCNs, the generator has the capacity to model complex relationship between FC and SC, which will be demonstrated by our experimental results in Section 4. Specifically, a generator that is composed by k multi-layer GCNs can be formulated by (2.6), (2.7) and (2.8),

$$G_i = G\left(\boldsymbol{T}, \boldsymbol{F}, \boldsymbol{W_i}\right) (2.6)$$

$$g\left(\left\{G_i\right\}, \boldsymbol{\theta}\right) = \boldsymbol{\theta} \oplus \left(G_1 \left\|G_2\right\| G_3\| \cdots \|G_k\right) = \sum_k \theta_k G_k (2.7)$$

$$\boldsymbol{T} = \begin{cases} g\left(\left\{G_i\right\}, \boldsymbol{\theta}\right), & \text{iteration } > 0 \\ \boldsymbol{F}, & \text{iteration } = 0' \end{cases} (2.8)$$

where $G_i, i = 1, 2, 3, \cdots, k$ represents the i^{th}GCN and $\|$ denotes parallel operation. Each GCN takes the individual FC($\boldsymbol{F}$) as input and outputs the predicted individual SC. Then, we used the learnable coefficients $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$ to fuse ($\oplus$) these k predictions and obtained the final prediction $\boldsymbol{S^p} = g\left(\left\{G_i\right\}, \boldsymbol{\theta}\right)$. During the training process, topology $\boldsymbol{T}$ is initialized by $\boldsymbol{F}$ and iteratively updated by $\boldsymbol{T} = g\left(\left\{G_i\right\}, \boldsymbol{\theta}\right)$. After training, each multi-layer GCN learns an independent mapping that represents a potential relationship between the input FC and SC. In order to enhance the capability of generator, we paralleled multiple GCNs to capture the complex relationships between individual SC and FC.

Single-GCN based Discriminator. In order to distinguish the two sets of graph data - real SCs and the predicted ones generated by the generator, the discriminator is composed by a multi-layer GCN, $G_d = G\left(SC, \boldsymbol{I}, \boldsymbol{W_d}\right)$, and followed by two fully-connected layers. The input SC can be the real SC matrix $-\boldsymbol{S}$, derived from diffusion MRI and predicted SC matrix $-\boldsymbol{S}^p$, created by generator. They are treated as real and fake samples during the training process. Different from generator, we used identity matrix as input feature matrix for discriminator. This is because discriminator aims to learn the rules that can be used to decide whether the input connectivity matrix is a valid SC matrix, any external knowledge should be excluded.

2.2.7 Structure-Preserving (SP) Loss Function

In the adversarial training scheme, the generator is optimized according to the feedback of discriminator. However, in this SC prediction task, the generator is trained to generate real-like individual SCs while the discriminator is trained to identify the real SCs from the predicted ones. The classification task of discriminator is much easier than the regression task of generator. Thus, the discriminator may easily differentiate real SCs from predicted SCs after a few training iterations and the generative adversarial loss would be close to 0 , resulting in zero back-propagated gradients in generator. In such case, the generator cannot be optimized and will keep generating invalid SCs. To break this dilemma, maintaining the balance between generator and discriminator regarding the optimization capability during the entire training process is important. We designed a new structure-preserving (SP) loss function to train our discriminator and generator. The SP loss function is combined by three parts: mean squared error (MSE) loss, Pearson's correlation coefficient (PCC) loss and GAN loss. It is formulated by (2.9), (2.10) and (2.11).

$$\mathcal{L}_{sp} = \mathcal{L}_{GAN} + \alpha\mathcal{L}_{MSE} + \beta\mathcal{L}_{PCC}, \alpha > 0, \beta > 0(2.9)$$

$$\mathcal{L}_{GAN} = -\left(\log(\mathcal{D}(\boldsymbol{S})) + \log\left(1 - \mathcal{D}\left(\boldsymbol{S^p}\right)\right)\right) - \log\mathcal{D}\left(\boldsymbol{S^p}\right) \tag{2.10}$$

$$\begin{aligned}
\mathcal{L}_{PCC} =& \mathcal{L}_{PCC-b} + \mathcal{L}_{PCC-r} \\
=& \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(s_{i,j} - \bar{s}\right)\left({s^{p^p}}_{i,j} - \overline{s^p}\right)}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(s_{i,j} - \bar{s}\right)^2}\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left({s^p}_{i,j} - \overline{s^p}\right)^2}} \\
&+ \sum_{i=1}^{n}\frac{\sum_{j=1}^{n}\left(s_{i,j} - \bar{s}\right)\left({s^p}_{i,j} - \overline{s^p}\right)}{\sqrt{\sum_{j=1}^{n}\left(s_{i,j} - \bar{s}\right)^2}\sqrt{\sum_{j=1}^{n}\left(s^{p_{i,j}} - \overline{s^p}\right)^2}} \\
& s_{i,j} \in \boldsymbol{S}, {s^p}_{i,j} \in \boldsymbol{S^p}
\end{aligned} \tag{2.11}$$

where the regularization parameters α and β are initialized by 1 and will gradually reduce to 0 later in the training process to let the model learn completely from the data. The three components of SP loss aim to guide the learning process from different perspectives. MSE loss ($\mathcal{L}_{MSE}$) forces the predicted SC to be the same scale as real SC at element-wise level. It is designed to control the magnitude of the predicted SC. PCC loss ($\mathcal{L}_{PCC}$) maximizes the similarity of overall pattern between predicted SC and real SC. It attempts to constrain the structure of the predicted SC. PCC loss is formulated by (2.11), which consists of two components: 1) brain-level PCC loss ($\mathcal{L}_{PCC-b}$) and 2) region-level PCC loss ($\mathcal{L}_{PCC-r}$). Brain-level PCC loss calculates the PCC between predicted SC matrix and real SC matrix, which measures the overall correlation between the predicted and real SCs. Region-level PCC loss calculates the correlation for each brain region (each row/column of the connectivity matrix), which measures the correlation of each brain region pairs of the predicted and real SCs. GAN loss ($\mathcal{L}_{GAN}$) effectively converts the regression problem to a classification problem and endows our model the power to implicitly learn the criterion, which is used to evaluate the quality of the predictions, from the data. It is formulated by (2.10), where $\mathcal{D}(\boldsymbol{S})$ and $\mathcal{D}\left(\boldsymbol{S}^p\right)$ are the classification results predicted by discriminator. The adversarial GAN loss guides the generator to create real-like SC to fool the discriminator by assigning a “true” label to the predicted SC as well as guides the discriminator to differentiate the two kinds of inputs correctly.

2.3 Results

We applied our proposed MGCN-GAN to infer individual SC from the associated FC. For each sample (subject) in training dataset, the real SC is used as the real sample for discriminator ((2.9) and (2.10)) as the ground truth for the generator at the beginning of the training process ((2.9) and (2.11)). The individual FC is used to initialize the adjacency matrix as well as to be the feature matrix ((2.6), (2.7) and (2.8)). During the adversarial training process, the topology of the graph is iteratively updated. The results of this work will be organized as follows: 2.3.1) introducing the experimental settings; 2.3.2) measuring the predicted SCs from different perspectives using two independent datasets; 2.3.3) evaluating the prediction performance of using different atlas; 2.3.4) evaluating different model settings including different GCN architecture, the learnable combination coefficients $-\boldsymbol{\theta}$, different loss functions; and 2.3.5) comparison with other widely used methods.

2.3.1 Experimental Setting

Data Setting. We conducted our experiments using two datasets: HCP and ADNI. For HCP dataset, we used 600 subjects for training and 464 subjects for testing. For ADNI dataset, we used 80CN subjects for training and 52CN subjects for testing. The details of the two datasets and the data preprocessing pipeline are introduced in Section 2.2.1 and 2.2.2. For each subject, following the process in Section 2.2.3, we created the individual SC and FC.

Model Setting. In this work, three two-layer GCNs are paralleled in generator. The model size of GCN components in generator is: $G_1 = (74, 148), G_2 = (148, 148)$

and $G_3 = (296, 148)$.$G_i = (F_1, F_2, \cdots, F_l)$ represents an l-layer GCN and output feature dimension of the lth layer is F_l. The three GCNs are combined by the learnable coefficient $\boldsymbol{\theta}$ which is initialized by $\boldsymbol{\theta}_i = (0, 0, 0)$. We also tested different model architectures and different initializations of $\boldsymbol{\theta}$ in Section 2.3.4. The discriminator is composed of one three-layer GCN followed by two fully connected layers. The model size of the GCN component is: $G_d = (148, 296, 148)$, and the output feature dimensions of the two fully connected layers are 1024 and 2, respectively. For both generator and discriminator, activation function Relu and layernorm are used at each layer. The entire model was trained in an end-to-end manner. During the training process, the Adam optimizer was used to train the whole model with standard learning rate 0.001 , weight decay 0.01 , and momentum rates $(0.9, 0.999)$.

2.3.2 Predicted Structural Connectivity

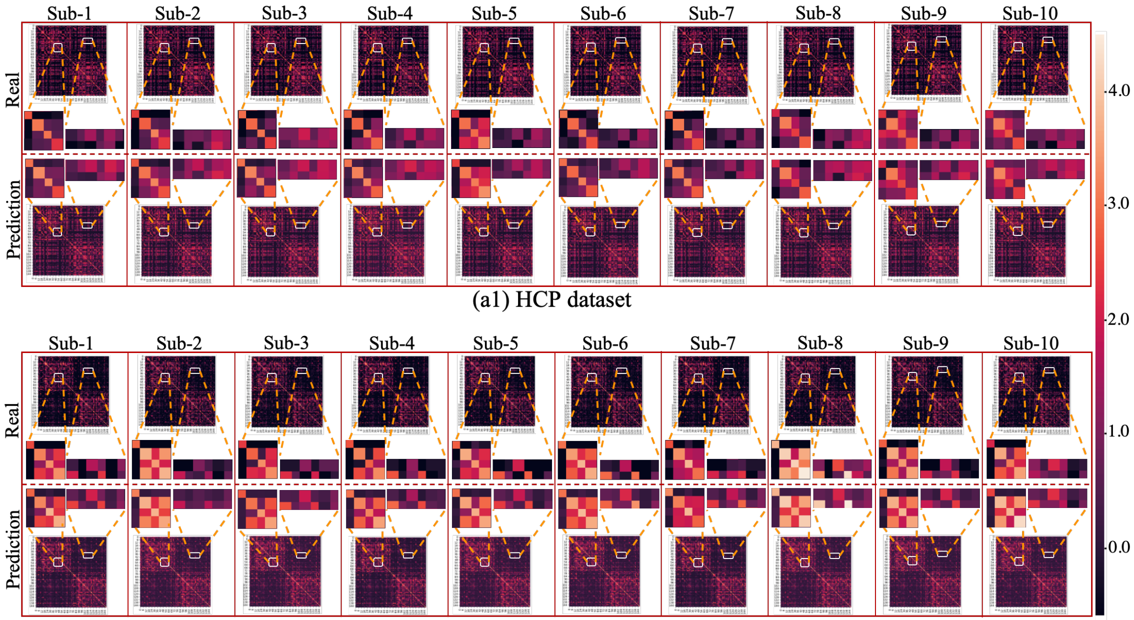
In this section, we evaluated the predicted SCs from three perspectives. Firstly, we visualized the predicted SC and real SC pairs to check the overall similarity upon visual inspection. Secondly, we quantitatively measured the similarity between predicted SCs with real ones from three ways (magnitude, overall pattern and graph property) via six measures (Mean Square Error (MSE), cosine similarity, Pearson correlation coefficient (PCC), degree, strength and clustering coefficient). Thirdly, we checked the prediction performance of predicted SC at connectivity level by comparing the top connectivity in predicted SCs and the real SCs. The individual SCs and FCs used in this section were generated via Destrieux atlas.

2.3.2.1 Visualization of predicted SCs and real SCs

To visually evaluate the similarity between the predicted SCs with the real SCs, we randomly selected 20 subjects from HCP dataset and ADNI dataset and showed the results in Fig. 2.3. We adopted two strategies to visualize the results. Firstly, we directly displyed the predicted SC and the real SC of each subject and showed the results in Fig. 2.3 (a1-a2). To demonstrate the details of the predicted results, we extracted two patches at the same location of predicted SCs and real SCs for all 20 subjects. These patches are enlarged and showed in the middle of (a1) and (a2). From the enlarged patches, we can see that different individuals possess subtle differences though their overall patterns are similar. Secondly, to better visualize the individual variability, for each of the forty matrices in (a1) and (a2), we displayed their corresponding connectivity matrix after subtracting the population-averaged SC and showed the results in Fig. 2.3 (b1-b2). We can see that after removing some consistent pattern across individuals by subtracting the averaged SC, the individual SC patten is much more obvious, and our predicted SC is highly similar to the real individual SC by visual examination. This result suggests that the predicted SC can effectively capture the individual variability. Of note, all these predictions are based on individual FC, which implies that there may exist a common regulation between specific brain structural and functional architectures across individuals.

2.3.2.2 Quantitatively measuring the similarity between predicted SCs with real SCs

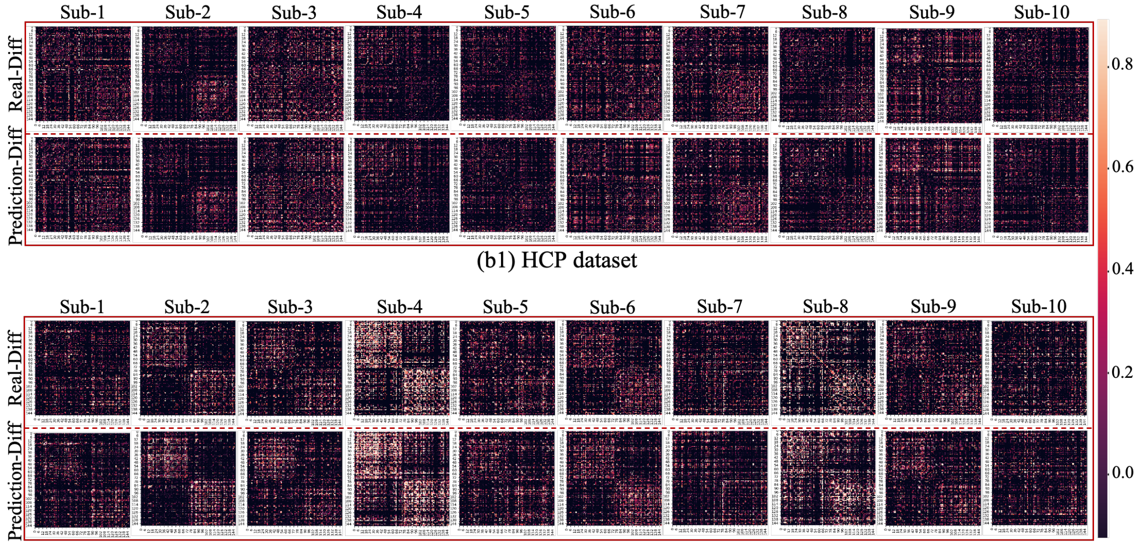
We quantitatively measured the similarity between predicted SCs and real SCs from three ways: magnitude, overall pattern and graph property. There are six measures in total, including Mean Square Error (MSE) for magnitude, cosine similarity and Pearson correlation coefficient (PCC) for overall pattern, and global metrics including mean degree, mean strength and mean clustering coefficient for graph property. In graph theory, the degree is the number of edges connected to a node and the mean degree is average of the degrees of all nodes, which is the most used measure of network density [57]. The strength of a node in a graph is defined as the increase in



(a1) HCP dataset

(a2) ADNI dataset

Real: Real SC generated by fiber count (*Section 3.3*) and normalized by formula (4) and (5)
Prediction: Predicted SC generated by proposed model
Real-Diff: Real SC minus Averaged SC of all Real SC
Prediction-Diff: Predicted SC minus Averaged SC of all Real SC

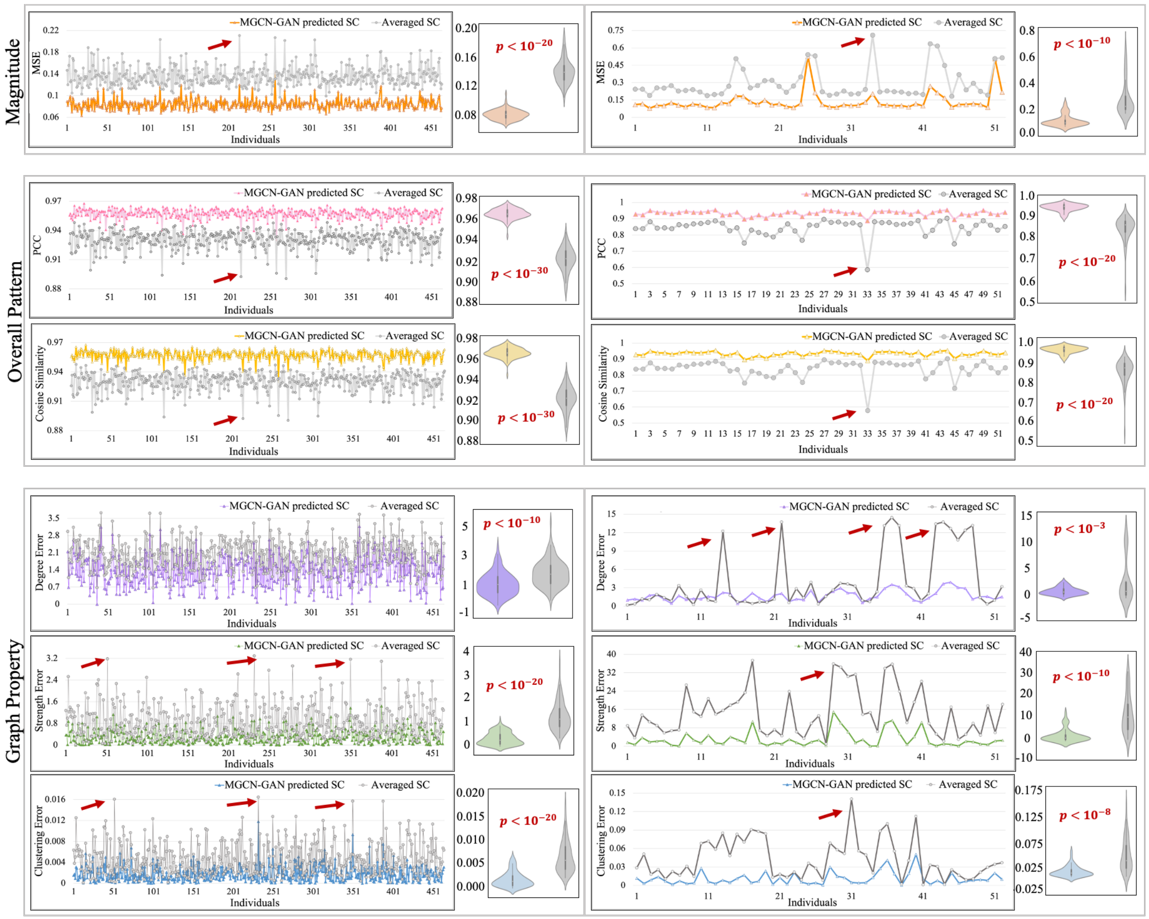


(b1) HCP dataset

(b2) ADNI dataset

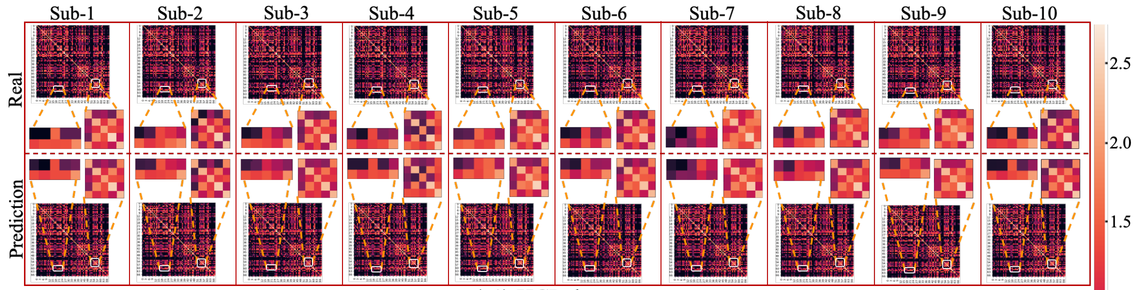
Figure 2.3. (a1-a2): Comparison of the predicted SCs and real SCs of 20 randomly selected subjects in HCP (a1) and ADNI (a2) datasets. For both datasets, we showed the 10 real SC matrices (the first row) and the corresponding 10 predicted SC matrices (the second row). Each column belongs to the same subject. Two patches of the matrices are extracted from the same location and their enlarged patches are showed in the middle. (b1-b2): Comparison of the predicted SCs and real individual SCs after subtracting the populaion-averaged SC. To better visualize the individual variability, the population-averaged SC was subtracting for each of the forty matrices in (a1) and (a2). The brain connectivity was generated via Destrieux atlas.

the number of connected components in the graph upon removal of the node, which measures the vulnerability of the graph and strictly generalizes the edge connectivity [58]. The clustering coefficient of a graph is the fraction of triangles around an individual node which is also equivalent to the fraction of the node's neighbors that are also neighbors of each other. The mean clustering coefficient for the graph hence reflects, on average, the prevalence of clustered connectivity around individual nodes [57]. All the three graph measures are widely used to describe the characteristics of a graph, such as segregation and integration [57]. For each subject, we calculated two MSE values, two cosine similarity values and two PCC values – one is between the prediction with the real SC and the other is between the averaged SC with the real SC, and two deviation values of each of the three global metrics – the deviation of the prediction from the real SC and the deviation of the averaged SC from the real SC. Finally, for each of the six measures, we obtained two sets of values. We showed the two sets of values by line chart and displayed the distributions by violin plot. We also performed significance analysis with p-value calculated via one tail two sample T-test. The results are shown in Fig. 2.4 for both of HCP dataset Fig. 2.4 (a) and ADNI dataset Fig. 2.4 (b). As shown in Fig. 2.4, 1) the predicted SCs have lower MSE, higher cosine similarity and PCC, smaller deviation of three global metrics and is significantly more accurate than the averaged SC; 2) the real SC of some subjects (highlighted by red arrows) show great deviation from the averaged SC, however, the predicted SCs of these subjects still can achieve remarkable performance, which suggests that our model can effectively characterize individual SC patterns; 3) the consistent results of two datasets indicate that the proposed model is stable across different datasets.

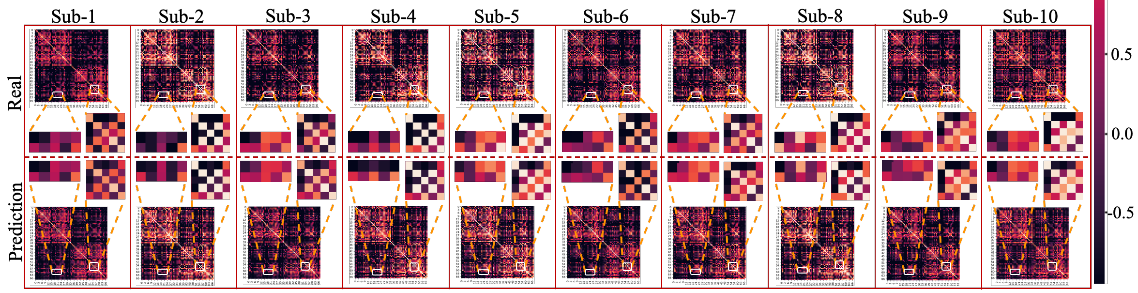


(a) HCP dataset (b) ADNI dataset

Figure 2.4. We quantitatively measured the similarity between predicted SCs with real SCs (based on Destrieux atlas) from three ways (magnitude, overall pattern and network property) by six measures (Mean Square Error (MSE), cosine similarity, Pearson correlation coefficient (PCC), mean degree, mean strength and mean clustering coefficient) of HCP (a) and ADNI (b) datasets. For each measure, two sets of values are calculated: one is the deviation of the predictions from the real SC and the other is the deviation of the averaged SC from the real SC. We showed the two sets of values by line chart and displayed the distributions by violin plot. The significance analysis was also conducted with p-value calculated by one tail two sample T-test.

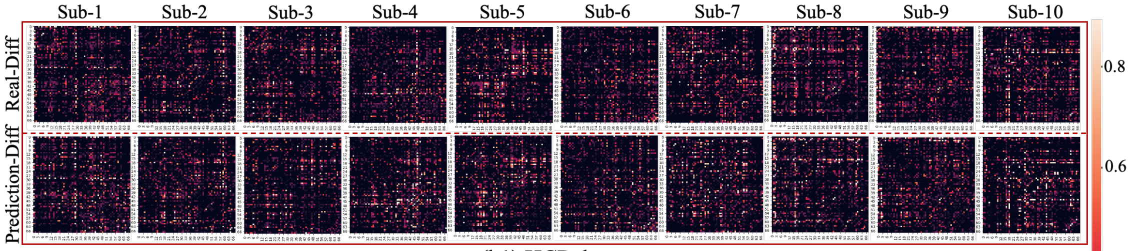


(a1) HCP dataset

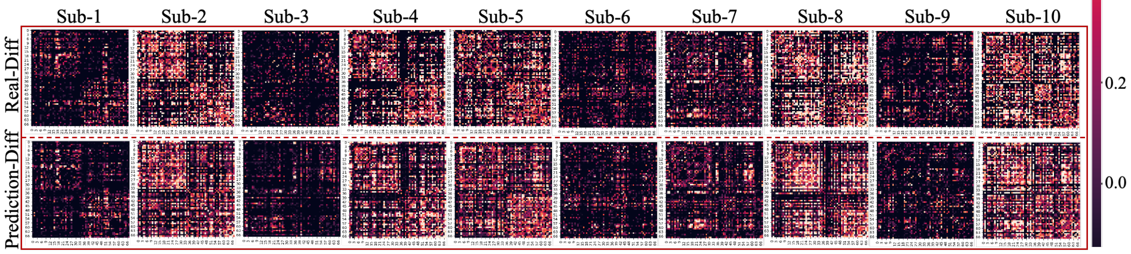


(a2) ADNI dataset

Real: Real SC generated by fiber count (*Section 3.3*) and normalized by formula (4) and (5)
Prediction: Predicted SC generated by proposed model
Real-Diff: Real SC minus Averaged SC of all Real SC
Prediction-Diff: Predicted SC minus Averaged SC of all Real SC

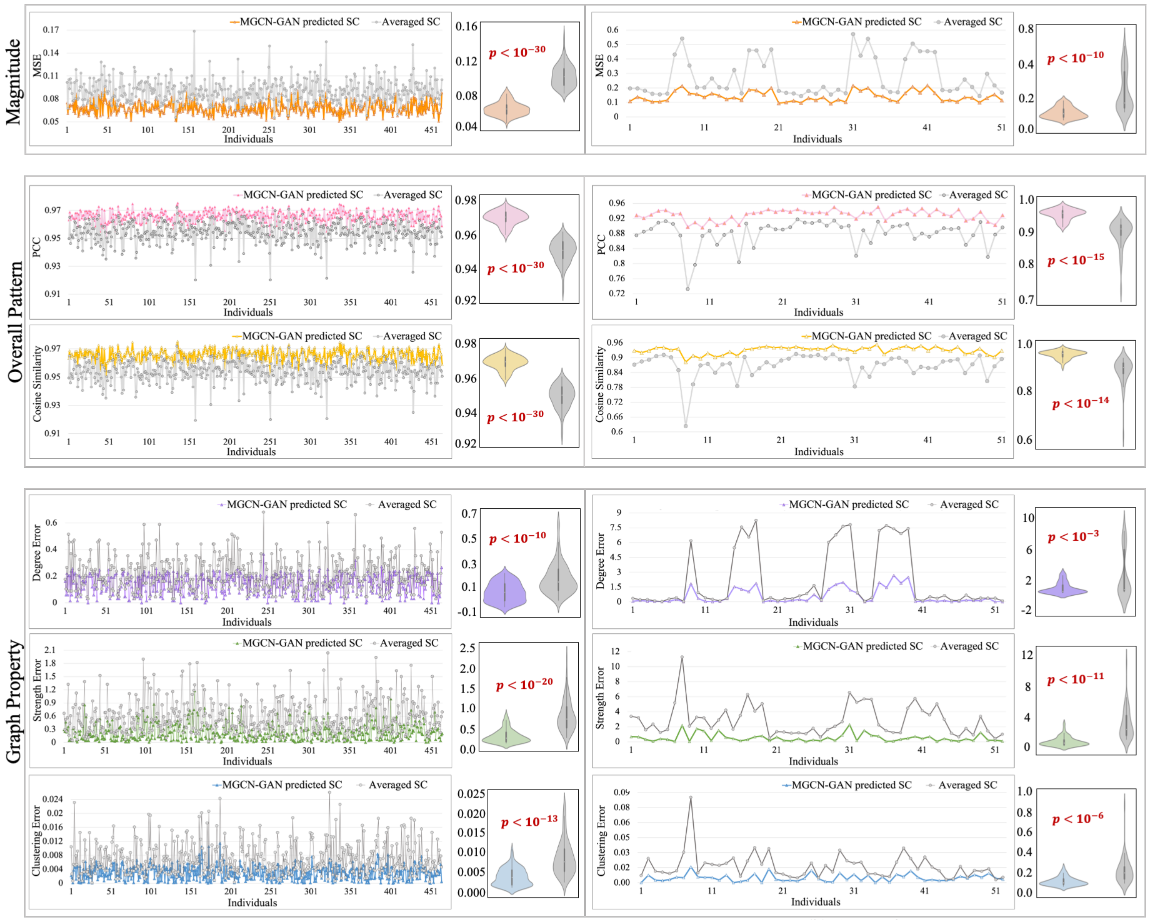


(b1) HCP dataset



(b2) ADNI dataset

Figure 2.5. (a1-a2): Comparison of the predicted SCs and real SCs of the same 20 subjects as used in Fig. 2.3. The brain connectivity was generated via Desikan-Killiany atlas. Each column belongs to the same subject. For each subject, we showed the real SC matrix in the first row and the predicted SC matrix in the second row. Two patches of the matrices are extracted from the same location and their enlarged patches are showed in the middle. (b1-b2): Comparison of the predicted SCs and real individual SCs after subtracting the population-averaged SC. To better visualize the individual variability, the population-averaged SC was subtracting for each of the forty matrices in (a1) and (a2).



(a) HCP dataset

(b) ADNI dataset

Figure 2.6. We quantitatively measured the similarity between the predicted SCs with real SCs (based on Desikan-Killiany atlas) from three ways (magnitude, overall pattern and network property) by six measures (Mean Square Error (MSE), cosine similarity, Pearson correlation coefficient (PCC), mean degree, mean strength and mean clustering coefficient) of HCP (a) and ADNI (b) datasets. For each measure, two sets of values are calculated, one is the deviation of the predictions from the real SC and the other is the deviation of the averaged SC from the real SC. We showed the two sets of values by line chart and displayed the distributions by violin plot. The significance analysis was also conducted with p-value calculated by one tail two sample T-test.

2.3.3 Evaluation the predicted SC using different atlases

The generation of brain connectivity relies heavily on the adopted brain atlas. To test the performance of the proposed model on different brain atlas, we used another widely used brain atlas - DesikanKilliany atlas, to generate individual SCs and FCs and conducted experiments. The predicted SC and the real SC based on Desikan-Killiany atlas of the same 20 subjects used in Fig. 2.3 were shown in Fig. 2.5. We can see that the results shown in Fig. 2.5 are highly consistent with that in Fig. 2.3 in the following aspects: 1) the real SCs of the 20 subjects also shows obvious individual variability, especially for the 10 subjects in ADNI dataset, which have significant variability in both magnitude and overall pattern. 2) for each subject, the prediction can capture the individual variability and obtain significant similarity with the real SC. This result suggests that the proposed model has good stability for different brain atlases.

To quantitatively measure the similarity between predicted SCs with real SCs based on Desikan-Killiany atlas, we calculated the MSE, cosine similarity, PCC, mean degree, mean strength and mean clustering coefficient for each subject in the testing dataset and showed the results in Fig. 2.6. As shown in Fig. 2.6, compared to the population averaged SC, the predicted SCs have lower MSE, higher cosine similarity and PCC, smaller deviation of three global metrics and can effectively characterize individual SC patterns. This result is consistent with that in Fig. 2.4, which further implies that the proposed model is general across different brain atlases.

2.3.4 Model Evaluation

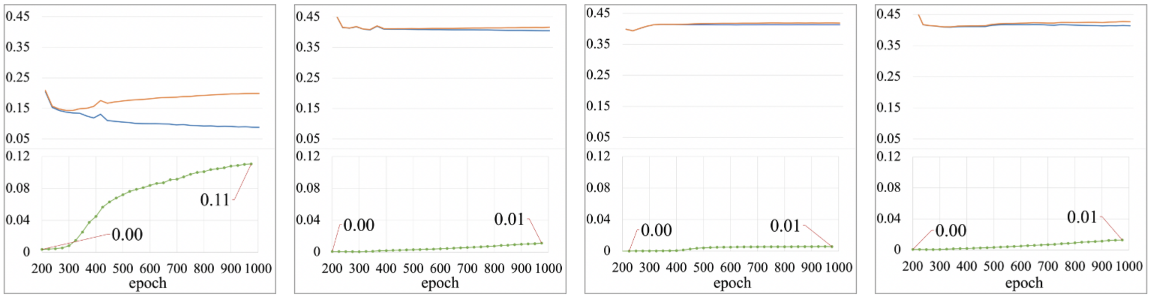
An effective model should have the capability to capture individual characteristics and avoid to being "trapped" in common SC patterns. To measure the effectiveness of a model, we proposed three measures:

1. MSE (Real, Prediction of same subject), which is the MSE between the real SC and predicted SC of the same subject. This measure directly evaluates the similarity between the real SC and the corresponding prediction. A smaller value indicates higher similarity. Thus, to generate reliable SC, this measure should keep decreasing before converged.
2. MSE (Real, Prediction of different subjects), which is the MSE between the prediction and the real SC of different subjects. A reliable prediction should avoid to being "trapped" in common SC patterns at population level. Therefore, this measure is expected to keep increasing during the training process.
3. MSE (Real, Prediction of different subjects) - MSE (Real, Prediction of same subject), which is the difference of the above two measures and an increasing value is expected.

In this section, using the three measures we evaluated different model settings including different GCN architecture, the learnable combination coefficients $-\boldsymbol{\theta}$, and different loss functions.

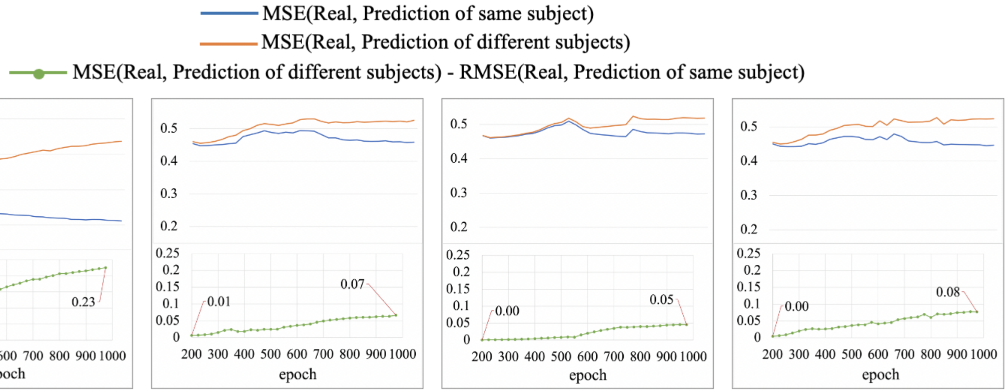
2.3.4.1 Evaluation of different GCN architectures

The generator was built on multiple GCNs, in order to verify the necessity of multi-GCN architecture, we conducted experiments to compare the performance of different generator architectures and showed the results in Fig. 2.7. We can see that, for predicted SCs generated from multi-GCN generator in both datasets (a1 and b1), the MSE (Real, Prediction of same subject) keeps decreasing and the MSE (Real, Prediction of different subjects) keeps increasing. For predicted SCs generated from single-GCN generator in both datasets (a2-a4, b2-b4), the difference between trajectories of MSE (Real, Prediction of same subject) and MSE (Real, Prediction of different subjects) is much smaller and the MSE (Real, Prediction of different



(a1) GCN(74||148||296) (a2) GCN(74) (a3) GCN(148) (a4) GCN(296)

(a) HCP Dataset



(b1) GCN(74||148||296) (b2) GCN(74) (b3) GCN(148) (b4) GCN(296)

(b) ADNI Dataset

Figure 2.7. Results of different generator architectures for HCP dataset (a) and ADNI dataset (b). GCN $(\mathrm{G}_1\|\mathrm{G}_2\|\ldots\|\mathrm{G}_\mathrm{k})$ represents the architecture of generator. The generator is composed of k two-layer GCNs, and the output feature dimension of the first layer of ith GCN is G_i.

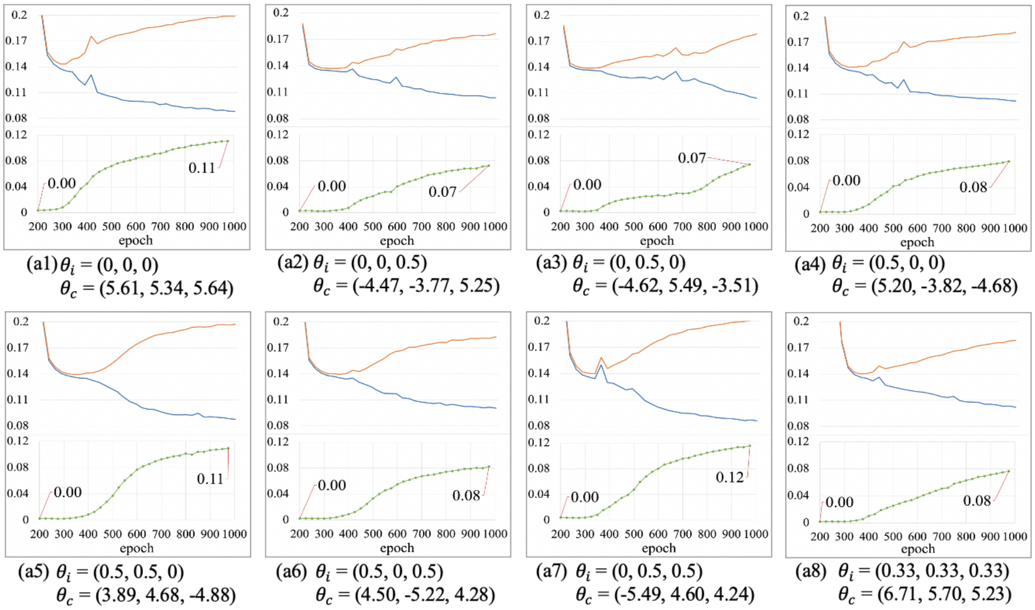
subjects) - MSE (Real, Prediction of same subject) only has slight increase. This result indicates that the predicted SCs generated from multi-GCN generator can efficiently learn the individual differences in SCs, while single-GCN generator only captures a common pattern at population level.

2.3.4.2 Evaluation of the learnable combination coefficients

In our model, the multiple GCNs in generator are combined by learnable coefficients $-\boldsymbol{\theta}$. In order to test the influence of the coefficients to the proposed MGCN-GAN model, we initialized the coefficients with different values and compared the prediction performance. The results are shown in Fig. 2.8. In general, the initialization of the learnable coefficients has very slight influence on the prediction results. Moreover, the coefficient with different initialization $\boldsymbol{\theta_i}$ always converge to stable coefficient $\boldsymbol{\theta_c}$ which is approximately equal for different GCNs in generator. It suggests all the GCNs have similar contributions to the results. Like the filters in CNN, multiple GCNs with different size of output features can be flexible and efficient for characterizing the complex FC-SC mapping.

2.3.4.3 Evaluation of SP loss function

To demonstrate the superiority of the proposed SP loss function, we compared our SP loss with GAN loss, combination of GAN loss and MSE loss, and combination of GAN loss and PCC loss and showed the results in Fig. 2.9. From the results we can see that the gap between trajectories of MSE (Real, Prediction of same subject) and MSE (Real, Prediction of different subjects) using SP loss function is increasing as the training progresses, which means individual characteristics are gradually learned. While the trajectories of MSE (Real, Prediction of same subject) and MSE (Real, Prediction of different subjects) using other three loss functions almost coincide during the training process and this implies that the other three loss functions may be limited in capturing potential subtle differences across individuals in the proposed model. The reason is that MSE only focuses on the element-wise similarity within the connectivity and overlooks the overall patterns. PCC has better performance in describing the overall connectivity patterns, but it may also overlook the connection magnitude across different connectivity and different individuals. However, both of MSE and PCC are important components in our designed SP loss to capture the subtle differences between real and predicted SCs.



(a1) θ_i = (0, 0, 0) θ_c = (5.61, 5.34, 5.64)

(a2) θ_i = (0, 0, 0.5) θ_c = (-4.47, -3.77, 5.25)

(a3) θ_i = (0, 0.5, 0) θ_c = (-4.62, 5.49, -3.51)

(a4) θ_i = (0.5, 0, 0) θ_c = (5.20, -3.82, -4.68)

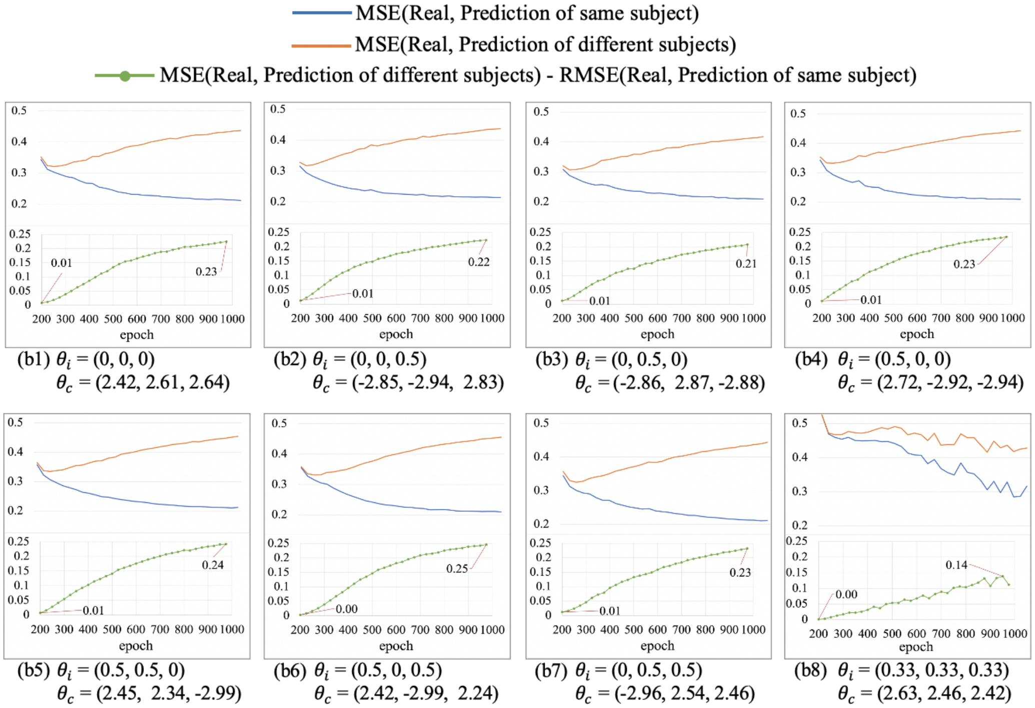
(a5) θ_i = (0.5, 0.5, 0) θ_c = (3.89, 4.68, -4.88)

(a6) θ_i = (0.5, 0, 0.5) θ_c = (4.50, -5.22, 4.28)

(a7) θ_i = (0, 0.5, 0.5) θ_c = (-5.49, 4.60, 4.24)

(a8) θ_i = (0.33, 0.33, 0.33) θ_c = (6.71, 5.70, 5.23)

(a) HCP Dataset



(b1) θ_i = (0, 0, 0) θ_c = (2.42, 2.61, 2.64)

(b2) θ_i = (0, 0, 0.5) θ_c = (-2.85, -2.94, 2.83)

(b3) θ_i = (0, 0.5, 0) θ_c =(-2.86, 2.87, -2.88)

(b4) θ_i = (0.5, 0, 0) θ_c = (2.72, -2.92, -2.94)

(b5) θ_i = (0.5, 0.5, 0) θ_c = (2.45, 2.34, -2.99)

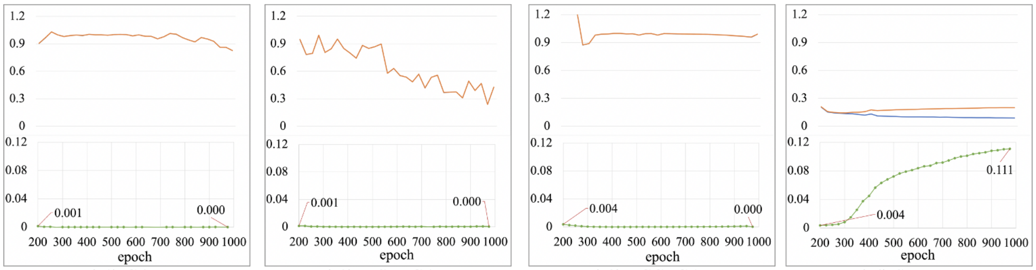
(b6) θ_i = (0.5, 0, 0.5) θ_c = (2.42, -2.99, 2.24)

(b7) θ_i = (0, 0.5, 0.5) θ_c = (-2.96, 2.54, 2.46)

(b8) θ_i = (0.33, 0.33, 0.33) θ_c = (2.63, 2.46, 2.42)

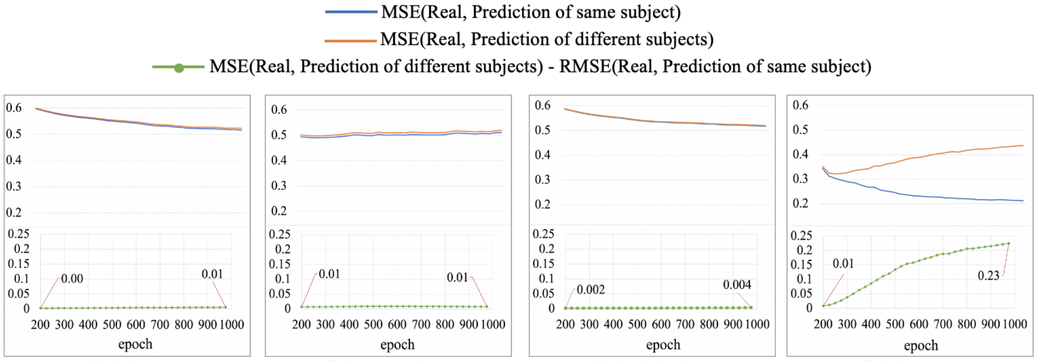
(b) ADNI Dataset

Figure 2.8. Results of different initializations of the learnable coefficients from HCP dataset (a) and ADNI dataset (b). $\boldsymbol{\theta_i}$ is the initialization of the learnable coefficients and $\boldsymbol{\theta_c}$ is the corresponding converged value.



(a1) GAN (a2) MSE+GAN (a3) PCC+GAN (a4) SP Loss

(a) HCP Dataset



(b1) GAN (b2) MSE+GAN (b3) PCC+GAN (b4) SP Loss

(b) ADNI Dataset

Figure 2.9. Results of MGCN-GAN with different loss functions on HCP dataset (a) and ADNI dataset (b).

2.3.5 Comparison with other widely used methods

To further demonstrate the effectiveness of the proposed MGCN-GAN, we compared the proposed model with three state-of-art models - CNN, multi-GCN, and CNN based GAN. In addition, for the comparison purpose we also included the linear regression into account. For fair comparison, we used the same dataset to train and evaluate the four methods (HCP dataset, 600 training/464 testing). Since Section 2.3.4 showed that both of MSE and PCC are important components to capture the subtle differences between real and predicted SCs, thus, we used MSE + PCC as loss function of linear regression, CNN, and Multi-GCN and used the proposed SP loss as loss function of CNN based GAN and the proposed MGCN-GAN. We used the six measures (Section 2.3.2) to evaluate the performance of different models and reported

Table 2.1. Comparison with other widely used methods.

Model Setting		Magnitude	Overall Pattern		Graph Property		
Methods	Loss Function	MSE	PCC	Cosine Similarity	Degree error	Strength error	Clustering Coefficient error
Linear regression	MSE+PCC	0.230±0.05	0.86±0.020	0.86±0.020	2.3±0.6	1.39±0.85	0.012±0.007
CNN	MSE+PCC	0.132±0.02	0.91±0.010	0.91±0.010	1.6±0.7	0.92±0.64	0.006±0.004
Multi-GCN	MSE+PCC	0.094±0.03	0.94±0.004	0.94±0.004	1.5±0.5	0.35±0.26	0.004±0.002
CNN based GAN	SP Loss	0.106±0.02	0.94±0.010	0.94±0.010	1.5±0.7	0.82±0.62	0.004±0.003
MGCN-GAN	SP Loss	0.084±0.01	0.96±0.005	0.96±0.005	1.3±0.6	0.29±0.25	0.002±0.001

the results in Table 2.1. As shown in Table 2.1: 1) compared to deep models, the performance of linear regression is much worse, which demonstrates the advantages of deep models in capturing complex nonlinear relationship between brain structural and functional networks; 2) the performance of GCN based models (Multi-GCN and MGCN-GAN) is better than CNN based models (CNN and CNN based GAN), and this result further suggests that GCN is a better choice to deal with the graph data; 3) comparing the performance between CNN and CNN based GAN and between Multi-GCN and MGCN-GAN, we can see, the GAN framework can effectively improve the performance of the model.

2.4 Discussion

2.4.1 Outliers in the experiments

In this work, we used six measures to quantitatively evaluate the similarity between predicted SCs and real SCs, including Mean Square Error (MSE) for magnitude, cosine similarity and Pearson correlation coefficient (PCC) for overall pattern, and global metrics including mean degree, mean strength and mean clustering coefficient for graph property. The results are shown in Fig. 2.4. From Fig. 2.4 we can see that there is a correlation between the two groups of MSE values, that is for some subjects which have a large MSE value between the population averaged SC and the real individual SC, the MSE value between the prediction SC and real individual SC is also slightly larger. Since all the samples we used in this work are normal brains, if a subject has significantly large MSE between individual SC and averaged SC comparing to other subjects, it is likely that this sample is an outlier. In such case, the MSE between the predicted SC and the real individual SC will be large, too. Therefore, the plot of the two groups of MSE values will show some correlated pattern. Even so, the difference between predicted SC and individual SC is much smaller than the difference between averaged SC and individual SC, this result implies our method is effective in characterizing the true relationship between SC and FC at individual level. Moreover, as this kind of correlation is not shared by other five measures, this correlated pattern should be related to the MSE measure.

2.4.2 Limitations and Future work

Spectral-based GCNs have become the most popular approaches for graph data recently because of their powerful ability to extract features from graph. Since the definition of orthonormal eigendecomposition of the Laplacian matrix $-\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ used in spectral-based GCNs depends on the adjacency matrix being symmetric [38], spectral-based models cannot directly work on directed graphs. In this paper, the input graph of GCNs is an undirected one, therefore this limitation has no influence on our results. Although in this work the proposed MGCN-GAN model is designed for the undirected graph, it is easy to extend the model to directed graph domain just by adjusting the definition of the adjacency matrix and the graph Laplacian without changing the architecture. For example, Monti et al. [59] divided a directed graph into a collection of small, directed graphs (graph motifs), then based on the graph motifs, they defined a symmetric motif adjacency and the motif Laplacian. Another widely used strategy is to define the in-degree matrix and out-degree matrix of a graph, respectively [60, 61], then based on the two matrices to construct Laplacian matrix and conducted graph convolution on directed graphs.

Another limitation of this work, which is also a general challenge suffered by deep neural networks, is the interpretability [62]. As defined in [63], interpretability is the degree to which an observer can understand the cause of a decision. The higher the interpretability of a model, the easier it is for observer to comprehend the problem, the data and the reason why a model might fail. Therefore, a variety of methods have been proposed to interpret neural network predictions. These methods can be divided into two major classes: 1) feature importance interpretation, which try to assign importance scores to each feature, including simple gradients method [64], DeepLift (Shrikumar et al. (2017)), and integrated gradients method [64]; and 2) sample importance interpretation, which assigns importance scores to each training example, such as influence functions method [65]. However, these methods cannot be directly applied to this work due to the following reasons: 1) this paper aim to study the brain structure-function relationship by recovering the individual SC from FC. Each input sample provides unique individual variability and are therefore equally important. Hence, sample importance interpretation is not suit for this work; 2) for feature importance interpretation, a common way is to generate saliency maps to highlight unique features which can depict the visually alluring locations in the input image. However, for non-Euclidean graph data, the important features are likely to be isolated points or small networks that are not continuous in space, which makes it difficult to distinguish them from noise. This further leads to a dilemma: the interpretation may be unexplanatory; and 3) Having an interpretation is not enough. The explanation itself must establish human trust. However, as suggested in [62], in most cases, for both feature-based and sample-based interpretation methods, systematic perturbations can lead to dramatically different interpretations without changing the label. Based on these reasons, currently there is no appropriate interpretation method for this work.

2.5 Conclusion

In this paper, we proposed a Multi-GCN based GAN (MGCN-GAN) model to generate individual SC from the corresponding individual FC. By adopting generative adversarial network (GAN), our proposed MGCN-GAN model can: 1) effectively handle brain's distributed and heterogeneous pattern; 2) learn the complex relationship between brain structure and function by leveraging adversarial training scheme to avoid designing an explicit regression loss function. By embedding multiple GCNs into GAN framework, our MGCN-GAN model can be used to represent the complex direct and/or indirect interactions in brain network. To overcome the inherent unstable behavior of vanilla GAN, we proposed a novel structure-preserving (SP) loss function to simultaneously capture the overall SC patterns and subtle differences across individuals during the training process. We tested our model and SP loss on two independent datasets (HCP and ADNI) and two different brain atlas (Destrieux Atlas and Desikan-Killiany Atlas, Section 2.3.3). The results demonstrate that our proposed model can effectively predict individual SC from the corresponding individual FC, and thus imply that there may exist a common regulation between specific brain structural and functional architectures across individuals.

CHAPTER 3

DEEP FUSION OF BRAIN STRUCTURE-FUNCTION IN MILD COGNITIVE IMPAIRMENT

In **Chapter 3**, we investigated ANNs in analyzing multi-modal imaging data for brain diseases study. Multimodal fusion of different types of neural image data provides an irreplaceable opportunity to take advantages of complementary cross-modal information that may only partially be contained in single modality. To jointly analyze multimodal data, deep neural networks can be especially useful because many studies have suggested that deep learning strategy is very efficient to reveal complex and non-linear relations buried in the data. However, most deep models, e.g., convolutional neural network and its numerous extensions, can only operate on regular Euclidean data like voxels in 3D MRI. The interrelated and hidden structures that beyond the grid neighbors, such as brain connectivity, may be overlooked. Moreover, how to effectively incorporate neuroscience knowledge into multimodal data fusion with a single deep framework is understudied. In this chapter, we developed a graph-based deep neural network to simultaneously model brain structure and function in Mild Cognitive Impairment (MCI): the topology of the graph is initialized using structural network (from diffusion MRI) and iteratively updated by incorporating functional information (from functional MRI) to maximize the capability of differentiating MCI patients from elderly normal controls. This resulted in a new connectome by exploring "deep relations" between brain structure and function in MCI patients and we named it as Deep Brain Connectome. The major advantage of the

Deep Connectome is its capability to integrate multiple types of network connectome and characterize their deep relationship as an “individual connectome signature”.

3.1 Introduction

With the availability of large-scale multiple types of brain image data, integration of data acquired from different imaging techniques, termed as multimodal data fusion [66, 67, 68, 69, 70, 71, 72], has gained considerable attention in neuroimaging field. Multimodal fusion provides an irreplaceable opportunity to take advantages of complementary cross-modal information that may only partially be contained in single modality data. Essentially, each imaging modality provides a different but unique view to represent brain structure and/or function. For example, diffusion tensor imaging (DTI) can provide information of brain structural connectivity (e.g., via tractography [73], and blood oxygen level-dependent (BOLD) signals derived from functional magnetic resonance imaging (fMRI) can be used to infer neural activity in vivo through measuring hemodynamic response. By jointly analyzing DTI and fMRI data we can investigate organizational architecture of human brain in both structural and functional domains. Besides studying general relationships between brain structure and function, multimodal data fusion can provide complementary knowledge when exploring and identifying potential abnormalities occurred in brain disorders. Mild cognitive impairment (MCI) is considered the precursor of Alzheimer’s disease, which is a progressive and irreversible neurodegenerative disorder characterized by severe cognitive decline and memory loss. Though the neuropathological mechanism is not fully understood, increasing evidences have shown that both structural and functional brain alterations are found in MCI patients [74, 75, 76, 77, 78, 79]. As a result, using single modality, e.g., either structural or functional data, for brain disease studies including classification or prediction, can be suboptimal.

Many multimodal fusion approaches have been published for cognitive and clinical studies [80, 19, 81, 82, 83, 84, 85, 86]. Most multimodal data fusion approaches have been focusing on simple and linear relationship between different modalities [87], while many studies have demonstrated the complexity between the information conveyed by different types of data: brain structure and function are closely related, but neither of them can fully interpret the organizational principles of human brain. Previous publications [19, 88] suggested that strong structural connectivity inclines to be accompanied with strong functional connectivity, but not vice versa. The changes of brain structure or function may also result in the alterations of the other. For example, when a structural connection between two brain regions is reduced, these regions can increase neuronal activity to compensate for the loss of communications [89].

As an emerging research topic, deep neural network based machine learning methods [32] can be especially useful in capturing the complex and non-linear relationship between brain structure and function in multimodal fusion. Recent development of deep learning has revolutionized the fields of artificial intelligence and machine learning [33], and boosted several domains such as computer vision [90] and many others [91, 92]. As one of the most popular deep models, convolutional neural network [8] (CNN) has demonstrated breakthroughs of performance in neuroimaging analysis including detection [36], reconstruction [34], segmentation [35] and computer-aided diagnosis [37]. However, it could be difficult to directly apply CNN on brain network analysis, as CNN and its numerous extensions are designed for operations on regular, Euclidean data, such as pixels in 2D slices or voxels in 3D volumetric data. Whereas human brain is a highly complex and interconnected network that can efficiently coordinate regional segregation and specialization [93, 94, 95]. The connectivity, which indicates the influences from remote brain areas that have either structural connections or functional interactions, provides the physiological basis of information representation and processing in brain [27, 96, 97, 98]. Therefore, the interrelated and hidden structures (i.e. modularity [94]) that beyond regular grid neighbors can be overlooked in traditional CNN approaches. In addition, most current deep neural network methods only focused on single modality data, such as structural, diffusion or functional MRI, very few studies have examined the multimodal data fusion using deep learning at network level. Though some deep models based on multimodal data have been proposed, the number and scope of studies are still very limited, and further efforts are highly needed to investigate appropriate strategies for exploring potential "deep relationship" between brain structure and function, as well as to develop disorder-focused deep modeling architecture.

Motivated by the merits of multimodal fusion in neuroimage studies and the recent advancement of graph convolutional network – GCN [38], we leverage both by integrating them into a novel graph-based deep model (GBDM) to study brain structure-function fusion at connectome level. We construct a multi-layer GCN with trainable graph topology. This graph is parameterized by both DTI-derived brain structural network and functional activities so that the learned graph becomes a deeply hybrid connectome by retaining brain structural substrate and simultaneously taking into account the functional influences as a complementary cross-modal information. In general, our GBDM aims to maximize the performance of classifying MCI patients from elder normal controls by incorporating functional interactions into structural network. Fig. 3.1 illustrates the overall strategy of GBDM. During the training phase, we used structural network as the initialization of the graph (Fig. 3.1(b)). Functional information (Fig. 3.1(a)) is used in two ways: the fMRI BOLD signals of each pair of brain regions are parameterized to form trainable functional profiles which are integrated into the current graph at each iteration; the functional connectivity are treated as features associated with the nodes that represent different brain regions. GBDM iteratively updates the topology of the graph (Fig. 3.1(c)) to minimize its classification loss. In the prediction phase, given individual structural network and functional data, GBDM outputs the graph topology as well as the predicted clinical conditions. Because this predicted graph evolves from both structural and functional connectome in a deep manner, we named it as ***Deep Brain Connectome***. Using ADNI (Alzheimer's Disease Neuroimaging Initiative) (http://adni.loni.usc.edu/) brain image dataset as a test bed, we generated deep brain connectome for each subject in both groups (MCI and aged normal controls). Compared to DTI derived structural connectome, the learned deep brain connectome displays decreased connectivity within the same hemisphere and increased connectivity across different hemispheres. These differences come from the integration of functional data when conducting classification task. Interestingly, the brain regions involved in the top changed connectivity are widely known for their close relations to AD/MCI development, such as lateral orbital and temporal regions. In addition, our experimental results show that the classification accuracy using our proposed GBDM yields up to 92.7%.

Our proposed deep multimodal fusion method advances the state of the art in three ways: first, a graph-based deep model is proposed to represent and manipulate brain connectome instead of using traditional CNN architecture which only considers the influences of Euclidean-based (local) neighbors. The primary motivation comes from the fact that one brain region might interact with other remote regions along the brain network topology. Many neurological symptoms are considered to be related to network level alternations rather than focal abnormality of brain [46, 99]. Second, the developed GBDM is trained based on structural network which is used to initialize the topology of the graph. The neurological rationality is that recent studies [100,

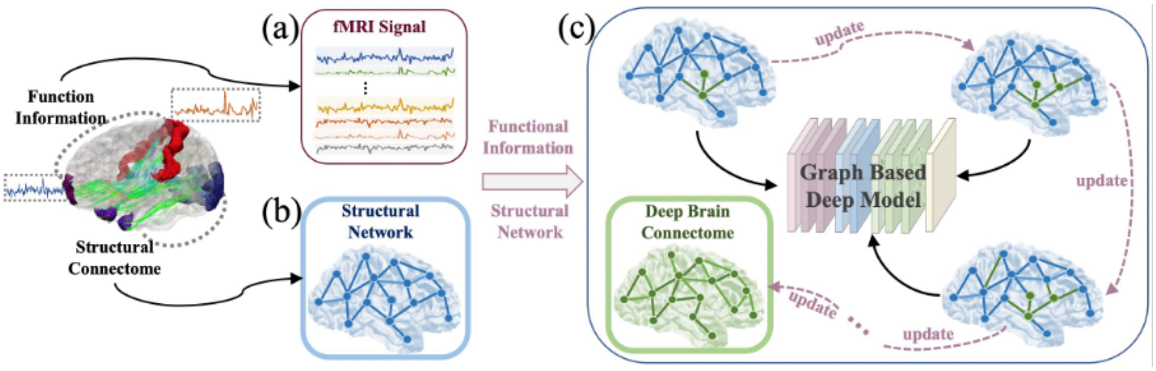


Figure 3.1. Illustration of the proposed Deep Brian Connectome learning based on Graph Based Deep Model (GBDM). By using Destrieux atlas [1] along with DTI and resting state fMRI data, we extracted the averaged BOLD signal of each brain region (148 regions in total) (a) and constructed brain structural network (b). Then the structural network was used as the initialization of graph topology, and the functional information was used by GBDM to iteratively update the topology of the graph – deep brain connectome (c), to maximize its classification power for MCI patients.

101] suggests that the neurodegenerative disease related pathogenic protein (e.g., Aβ and tau) may aggregate via long-distance transmission along white matter pathways between remote brain regions. Hence, it is natural to explore potential MCI/AD related abnormalities within individual structural network, instead of using Euclidean neighbors. Thirdly, most current deep learning methods focused on either brain structural or functional data, few studies have been conducted on both of them at network level. However, it is suggested that MCI/AD progression may follow both structural and functional topologies [78]. In our proposed GBDM, the topology of the graph is learned from functional features and structural network simultaneously, by maximizing MCI classification performance. The obtained deep brain connectome reflects a deep fusion of structural and functional network alterations in MCI patients.

3.2 Materials and Methods

3.2.1 Methods Overview

We proposed a graph-based deep model (GBDM) (Fig. 3.2) to analyze brain structure-function abnormalities in MCI patients by integrating both structural and functional data. Brain structural network is used to initialize the topology of the graph, i.e., the adjacency matrix in GCN. An individual functional profile is learned and combined with structural network iteratively. Pearson's correlation coefficients of averaged BOLD signals are treated as the features associated with the nodes of the graph. The entire model is designed to differentiate MCI from NC by seamlessly incorporating functional profile learning (Section 3.2.2) and brain structure-function fusion (Section 3.2.3). Specifically, our model is composed of four components: 1) learning of functional profile to parameterize pairwise functional relations between any two brain regions; 2) brain structure-function fusion for seeking to best combine both structural network and the learned functional profile as new topology of the graph; 3) brain network convolution conducted upon the updated graph topology (Section 3.2.4); 4) MCI-NC classification with fully connected neural network (Section 3.3.4).

3.2.2 Functional Profile Learning

There exist a few measurements to represent pairwise relationship between two BOLD signals derived from fMRI, such as mutual information [102], covariance [52], correlation [46] and partial correlation [103]. In general, how to effectively represent the functional relationships among brain regions is still an open research area. In this work, we aim to learn a disease-related functional mapping matrix to form a functional profile that can be used to combine with brain structural network at later step. We will parameterize the representation of the similarity between two BOLD signals and the parameter will be automatically learned during the training process. The steps of the parameterization are presented as follows:

Averaged fMRI Signal Extraction: We used the Destrieux atlas [1] to divide the whole brain into 148 regions. We calculated the average fMRI signal for each brain region as the representative for later analysis.

Normalization: We normalized the averaged fMRI signal using the standard Z-score normalization [50] shown in (3.1),

$$f_i^* = \frac{f_i - f_\mu}{f_\sigma}(3.1)$$

where f_i is the averaged fMRI signal of brain region i, f_μ and f_σ are the mean and the standard deviation of all 148 averaged fMRI signals.

Functional Profile Learning: We defined the parameterized functional-pairwise distance between region- i and region- j using (3.2):

$$\phi\left(f_i^*, f_j^*; M\right) = \|M\left(f_i^* - f_j^*\right)\|^2, \forall i, j \varepsilon 1, \ldots, N(3.2)$$

where $M \in R^{T \times T}$ is the learnable functional mapping matrix, T is the number of time points of averaged fMRI signals. f_i^* and f_j^* are the normalized averaged signals of two brain regions. It is worth noting that the proposed parameterization approach (3.2) has been widely used in metric learning, where M is interpreted as a linear projection matrix and $\phi\left(f_i^*, f_j^*; M\right)$ becomes the Mahalanobis distance with precision matrix $M^T M$ [104]. Different from conventional metric learning methods, we will integrate it into the proposed deep learning model with a natural nonlinear extension. Due to the high complexity of fMRI signals, linear projection function used in (3.2) is inadequate for modeling the distance/similarity of the averaged fMRI signals. To alleviate the above issue, nonlinear projection can be introduced by applying Gaussian kernel in terms of kernel trick given by

$$A_{ij}^{F}=\exp\left(-\frac{\phi\left(f_i^*,f_j^*;M\right)}{2\sigma^2}\right),\forall i,j\in 1,\ldots,N(3.3)$$

where $A_{ij}^{F}\in R$ represents the pairwise functional profile between brain region i and region $j.A^{F}\in R^{N\times N}$ is the matrix including all pairwise functional profiles between any two brain regions with the $(i,j)^{\text{th}}$ entry – A_{ij}^{F},σ is the bandwidth parameter of Gaussian kernel and is treated as a hyper-parameter, which will be fully discussed in Section 3.3. In order to avoid introducing any bias, M is initialized as identity matrix. During the training process, M is iteratively updated based on classification results.

3.2.3 Brain Structure-Function Fusion

A major goal of this work is to examine MCI related network alterations via deep fusion of brain structural and functional data. To this end, we propose a novel strategy to integrate brain structural networks and functional profiles in a deep and regularized fashion. First, we calculate individual structural network matrix $A^{S}\in R^{N\times N}$, which is a symmetric matrix and $A_{ij}^{S}\in R$ is the number of DTI-derived fibers connecting brain region i and region j. Then, we conduct normalization of A^{S} using (3.4) and (3.5). $\left(A^{S}\right)_{\mu}$ and $\left(A^{S}\right)_{\sigma}$ are the mean and the standard deviation of A^{S}. Because the number of fibers connecting two brain regions can be from zero to a few thousands and conform to a skewed distribution. Log transformation can equalize the standard deviations and make the distribution of the sample mean more consistent with a normal distribution (Curran-Everett (2018)). Therefore, we first used log transformation to narrow the range of the number of fibers by (3.4) and then used (3.5) for normalization.

$$A_{ij}^{S}=\log_{10}\left(A_{ij}^{S}+1\right)(3.4)$$

$$A^S = \frac{A^S - \left(A^S\right)_\mu}{\left(A^S\right)_\sigma} (3.5)$$

At last, we combine the normalized structural network $\left(A^S\right)$ and functional profile $\left(A^F\ \right)$ using (3.6) and (3.7):

$$\hat{A} = I + \theta_1 A^S + \theta_2 A^F (3.6)$$

$$\theta_k = \frac{\exp\left(-\beta_k\right)}{\sum_{m=1}^{2} \exp\left(-\beta_m\right)} (3.7)$$

where I is an identity matrix of $N \times N$ and it makes sure that each node can be considered in its own convolutional operations. θ_1 and θ_2 control the contributions of structural and functional components in the combined new brain connectome ($\hat{A}$). Here $\beta_k (k = 1, 2)$ are introduced in (7) to constrain $\sum_{k=1}^{2} \theta_k = 1$ and $\theta_k > 0$. During this training process, θ_k is iteratively updated (via β_k) to improve the classification performance when differentiating MCI from NC in the fully connected layers (Section 3.5). It is clear that introducing β_k s in (7) can facilitate the gradient descend method to optimize $\theta_k s$ without cautious on the simplex constraint, and (7) is a continuous and differentiable function, which can be easily optimized by the backpropagation method. As a result, the presented fusion strategy is easily incorporated into our deep learning models. In general, the disease-related knowledge (from classification) is passed to functional profile $\left(A^F\right)$ and then transferred to the new brain connectome ($\hat{A}$), by combining with structural network $\left(A^S\right)$. This is an iterative process and at each iteration, $\hat{A}$ will be used as the new topology for graph convolution of node-associated features.

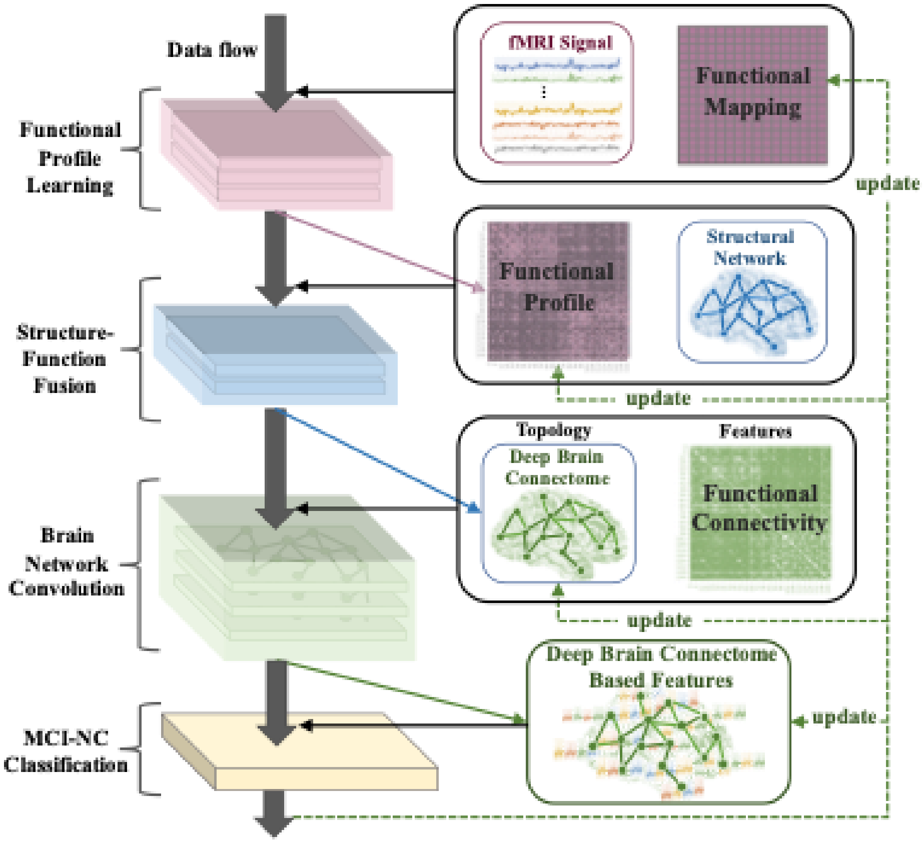


Figure 3.2. The major steps of the proposed Graph Based Deep Model (GBDM). The model consists of four components: 1) functional profile learning, 2) brain structure-function fusion, 3) brain network convolution and 4) MCI-NC classification. Functional profile learning aims to learn a new representation of pairwise functional relations by introducing a trainable functional mapping matrix. In the step of brain structure-function fusion, structural network and the learned functional profile are combined as a hybrid brain network and it will be updated in each iteration. Using functional connectivity (defined with Pearson's correlation coefficient) as features associated to the network nodes (brain regions), we conduct graph convolution based on the hybrid brain network. A fully connected network is appended at last for MCI-NC classification and the results are used to update the parameters in previous layers. Because the learned hybrid brain network "deeply" integrates brain structure-function information to maximize its MCI-NC classification power, we name it as Deep Brain Connectome.

3.2.4 Brain Network Convolution

To represent the latent interactions in brain network, we adopt a multi-layer GCN architecture to fulfill the joint tasks of classification and brain connectome learning. GCN [38] extends traditional CNN by applying convolutional operations on graph-based instead of Euclidean-based neighbors. Here we provide a basic definition of graph for better understanding of GCN.

Definition. A graph is denoted by $G = (N, E, A)$ where N is the set of nodes, E is the set of edges and A $\in R^{N\times N}$ is the adjacency matrix, N is the number of the nodes. In a graph, let $n_i \in$ N denote node $i, e_{ij} = (n_i, n_j) \in$ E denote an edge connecting n_i and n_j and $a_{ij} \in$ A to represent the connection strength between n_i and n_j.

For a graph G, each node n_i can have its own attributes represented by a row vector $x_i \in R^{1\times D}$ and D is the dimension of attributes (features). X $= [x_1, \ldots, x_N] \in R^{N\times D}$ is the feature matrix of

the graph. We used the updated hybrid brain connectome $\hat{A}$ as the adjacency matrix in each iteration (A $= \hat{A}$). We calculated Pearson Correlation Coefficient of every pair of averaged fMRI signals as feature matrix $P \in R^{N\times N}$(X $= P$). The feature vector of node n_i, $P_i = (P_{i1}, P_{i2}, \ldots, P_{iN})$, is the concatenation of correlations to all the other nodes. The input graphs in our model are individuallevel graphs which have the same number of nodes representing the corresponding brain regions. More importantly, the topology of the graphs (A $= \hat{A}$) are flexible and will be iteratively updated in the training process to maximize its classification power when differentiating MCI from NC.

Based on P and $\hat{A}$, the convolutional process of multi-layer graph convolutional network is formulated by (3.8) and (3.9).

$$G = f\left(\hat{A}H^{l-1}W_l\right) \tag{3.8}$$

$$H^l = \begin{cases} f\left(\hat{A}H^{l-1}W_l\right), l \geq 1, \\ P, l = 0, \end{cases} \tag{3.9}$$

where f is the nonlinear activation function and we used Relu in our experiment. For the tth layer, H^l is the output, $W_l \in R^{F_{li} \times F_{lo}}$ is the weight matrix, F_{li} and F_{lo} are the feature sizes of input and output feature matrix. As showed in Fig. 3.3, each W_l is a filter which selects related features from neighbors and defines how to combine these features. By stacking multiple graph convolutional layers, information from high-order neighbors (indirectly connected via other nodes) can be propagated along brain connectome defined by current adjacency matrix $\hat{A}$. Fig. 3.4 illustrates the neighbors with different order and how a multi-layer GCN conducts the graph convolution at different layers. Our previous study [105] suggested that the impact of MCI-related alterations may go beyond the first-order neighbors and be limited to the third-order neighbors. Therefore, in our experiments, brain network based convolutions were implemented using a two-layer GCN.

3.2.5 MCI-NC Classification

In the last part of GBDM, we designed a two-layer fully connected neural network to perform binary classification for two classes - MCI and NC. Based on the classification performance, the gradient information will be propagated back to the entire network and the following parameters will be updated accordingly: 1) $\{W_l\}$ in brain network convolution - Section 3.2.4 ; 2) $\{\theta_k\}$ in brain structure-function fusion

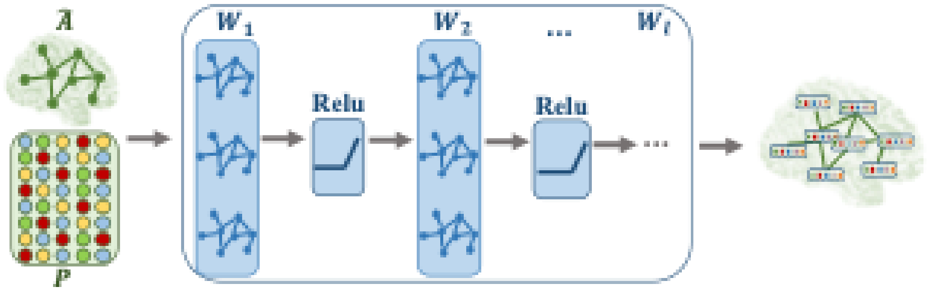


Figure 3.3. Multi-layer graph convolutional network. According to the input adjacency matrix $\hat{A}$ and feature matrix P, a GCN layer creates a hidden representation for each node by combining features from its neighbor nodes based on W_l. After the combination, a nonlinear transformation is applied to the hidden representation. By stacking multiple layers, the final hidden representation of each node gathers information from both direct neighbors and high-order neighbors (indirectly connected via other nodes).

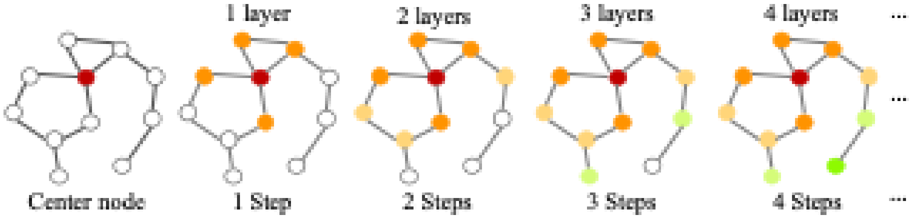


Figure 3.4. The influences of using different numbers of GCN layers for aggregating features. We use step to represent the number of edges in a shortest path connecting two nodes in the graph. The nodes with the same color represent that they have the same steps from the center node (red). With more GCN layers are stacked the deeper of brain network relations are considered.

- Section 3.2.3; and 3) M in functional profile learning - Section 3.2.2. Through this way, the disease-related knowledge drives the training process to learn a new brain connectome - $\hat{A}$ (see (3.6)) which represents a deep fusion of brain structure and function, that is the deep brain connectome. After the above three parameters are obtained after training process, we can make prediction for any given individual data including structural network matrix A^S, feature matrix P, and averaged fMRI signals $\{f_i\}$: the first step is to calculate the individual deep brain connectome $\hat{A}$ by using (3.6) and the optimized M and $\{\theta_k\}$; then P together with $\hat{A}$ are forward propagated through the trained deep graph convolutional network with optimized $\{W_l\}$ to the classification layer for final prediction.

3.3 Results

We applied our proposed GBDM to individual fMRI signals and brain structural network. For each sample (subject) in training data, the individual structural network is used to initialize the adjacency matrix with (3.6). Individual functional signals are used for functional profile learning (see (3.3)), brain structure-function fusion (see (3.6)) as well as node features in (3.9). Besides classification, the major outcome of GBDM is the learned hybrid connectome - deep brain connectome ($\hat{A}$), which integrates the knowledge from both brain structural and functional data. In the results, Section 3.3.1 and 3.3.2 introduce the participants and data preprocessing. Section 3.3.3 shows the experimental setting. Section 3.3.4 shows the comparison of classification performance with other widely used methods. Section 3.3.5 shows the details of the learned deep brain connectome and the evaluation of different model settings will be discussed in Section 3.3.6.

3.3.1 Participants

In this work, we jointly analyzed three modalities of brain imaging data in ADNI dataset, including structure MRI (T1weighted), resting state fMRI (rs-fMRI) and DTI. We began with 252 subjects (132 Normal Controls (NC) and 120MCI patients) which have all three modalities. 38 (16 NC and 22MCl) subjects were not used in this work due to poor image quality. In total, we have 214 subjects including 116 subjects from NC group (60 females, 56 males; 74.26 ± 8.42 yrs.) and 98 subjects from MCI group (57 females, 41 males; 74.23 ± 8.81 yrs.). The proposed analysis was conducted on these 214 subjects.

3.3.2 Data Description and Preprocessing

The T1-weighted MRI data has $240 \times 256 \times 208$ voxels and the voxel size $= 1.0\ \text{mm} \times 1.0\ \text{mm} \times 1.0\ \text{mm}, \text{TR} = 2.3$ s. The DTI data has 54 gradient directions, each volume has $116 \times 116 \times 80$ voxels and the voxel size $= 2.0\ \text{mm} \times 2.0\ \text{mm} \times 2.0\ \text{mm}, \text{TE} = 56$ ms and $\text{TR} = 7.2$ s. The rs-fMRI data has 197 volumes, each volume has $64 \times 64 \times 48$ voxels and the voxel size $= 3.4375\ \text{mm} \times 3.4375\ \text{mm} \times 3.4\ \text{mm}, \text{TE} = 30\ \text{ms}, \text{TR} = 3$ s and flip angle $= 90°$. The first 6 volumes were discarded during preprocessing procedures to ensure magnetization equilibrium.

We applied the same standard preprocessing procedures as in [46]. In brief, we applied skull removal for all modalities. For rs-fMRI images, we applied spatial smoothing, slice time correction, temporal pre-whitening, global drift removal and band pass filtering (0.01-0.1 Hz). All of these preprocessing steps are implemented using FMRIB Software Library (FSL) [47] FEAT. For DTI images, we applied eddy current correction via FSL and fiber tracking via MedINRIA. For T1-weighted images, we registered them to DTI space by FSL FLIRT and then conducted segmentation using FreeSurfer package [49]. After the segmentation, we adopted the Destrieux Atlas [1] for ROI labeling and the brain cortex is partitioned into 148 regions after removing two unknown areas and two empty areas.

3.3.3 Experimental Setting

Data Setting. For each subject, the whole brain is divided into 148 regions using Destrieux Atlas. We calculate averaged fMRI signal for each brain region and created brain structural network $\left(A^S\right)$ and Pearson Correlation Coefficient matrix (P) for each subject. For classification, we conducted 5-fold cross-validation using the 214 subjects (116NC/98MCI).

Model Setting. The functional profile learning was implemented by one fully connected layer with the input dimension and the output dimension of T, where T is the number of time points. We employed a two-layer GCN for brain network convolution. The output feature dimensions of the first GCN layer and the second GCN layer are 148 and 296, respectively. The two feature dimensions are selected according to model evaluation (Section 4.6). The MCINC classification was conducted by one fully connected layer with input dimension 43808 (region number (148) × feature size (296)) and output dimension C, where C is the number of classes ($C = 2$ in this work). The entire model was trained in an end-to-end manner. During the training process, the parameters were initialized following the Xavier scheme. The Adam optimizer was used to train the whole model with standard learning rate 0.0005 , weight decay 0.01 , and momentum rates $(0.9, 0.999)$.

3.3.4 Classification Performance

The proposed GBDM is based on MCI/NC classification task. In this section, we focus on comparing classification performance of the proposed method with other widely used methods. As mentioned earlier, GBDM is designed to learn a deeply combined structural-functional connectome that can be used to achieve higher MCI/NC classification performance. For fairly comparison, we summarize the overall classification performance of recent MCI studies using both single-modality and multi-modality as well as our model in Table 3.1. The approaches of classification include traditional machine learning methods, such as SVM [106, 107, 108, 109, 110, 111] and most recent deep learning models, such as CNN [112, 86, 113]. The single modality includes MRI, DTI, and PET. The multi-modality includes MRI + PET, MRI + PET + biomarkers, MRI + PET + biomarkers + Genetics, MRI + DTI, and MRI + DTI + fMRI. For AD/NC classification, the accuracy of most single-modality methods listed in Table 1 is bellow 90% while the accuracy of most multi-modality methods is over 90%. Although the performance of [106] and [109] stand out than other listed single-modality methods and can reach an accuracy over 90%, they used multi-view and multi-atlas to organize the single-modality data at different scales and different views. For MCI/NC classification, the accuracy of most single-modality methods is bellow 80% while the accuracy of most multi-modality methods is over 80%. Among the listed studies, [112], [86] and [111] tested the proposed methods with both single-modality and multi-modality. Their results indicate that comparing to single-modality, using multi-modality data can achieve higher classification accuracy. Some of the listed multi-modality methods used PET data, which inclines to have better classification power than MRI based modalities. It is worth noting that our method used noninvasive DTI/fMRI data and achieves the best performance of 92.7% for MCI/NC classification. Our result suggests that recognition of the complex relationship between different modalities may be critical for better understanding brain structure-function alterations in brain disorder studies.

3.3.5 Deep Brain Connectome

Besides classification of MCI patients, the other major outcome of GBDM is the learned deep brain connectome ($\hat{A}$). By comparing the learned deep brain connectome with the original brain structural network, we are able to examine the functional influences on the structural connectivity (fusion of structural and functional data) when conducting MCI classification task. All the results showed in this section are based on the testing dataset.

Table 3.1. Performance Comparison of Different Methods in AD/MCI Classification.

(A) Single-Modality Study

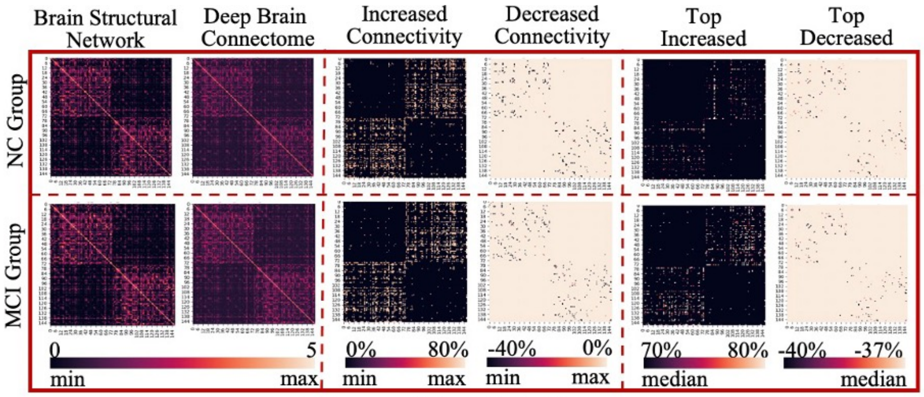
Study	Modality	Group			Method	Accuracy (%)			
		AD	MCI	NC		AD/ NC	MCI/ NC	AD/ MCI	pMCI/ sMCI
Apostolova et al. (2014)	MRI	95	182	111	Hippocampus; SVM	85.0	79.0	70.0	-
Min et al. [106]	MRI	97	117^p+117^s	128	Multi-atlas; Morphometry; SVM	91.6	-	-	72.4
Liu et al. [109]	MRI	97	117^p+117^s	128	multi-view; multi-template; SVM	93.8	-	-	80.9
Möller et al. [110]	MRI	84	-	94	Gray matter density maps; SVM	88.0	-	-	-
Aderghal et al. [112]	MRI	188	399	228	Hippocampus; CNN; transfer learning	90	72.5	82.5	-
Aderghal et al [112]	DTI	48	108	58	Hippocampus; CNN; transfer learning	85	62.5	82.5	-
Liu et al. [86]	MRI	93	76^p+128^s	100	Patch-based; Cascaded CNNs	85.0	77.8^p 60.1^s	-	-
Liu et al. [86]	PET	93	76^p+128^s	100	Patch-based; Cascaded CNNs	88.1	78.4^p 63.4^s	-	-
Shao et al. [111]	MRI	160	273^e+187^l	160	Hypergraph; Multi-kernel SVM	88.3	69.1	-	68.5
Shao et al. [111]	PET	160	273^e+187^l	160	Hypergraph; Multi-kernel SVM	87.3	65.6	-	67.6

(B) Multi-Modality Study

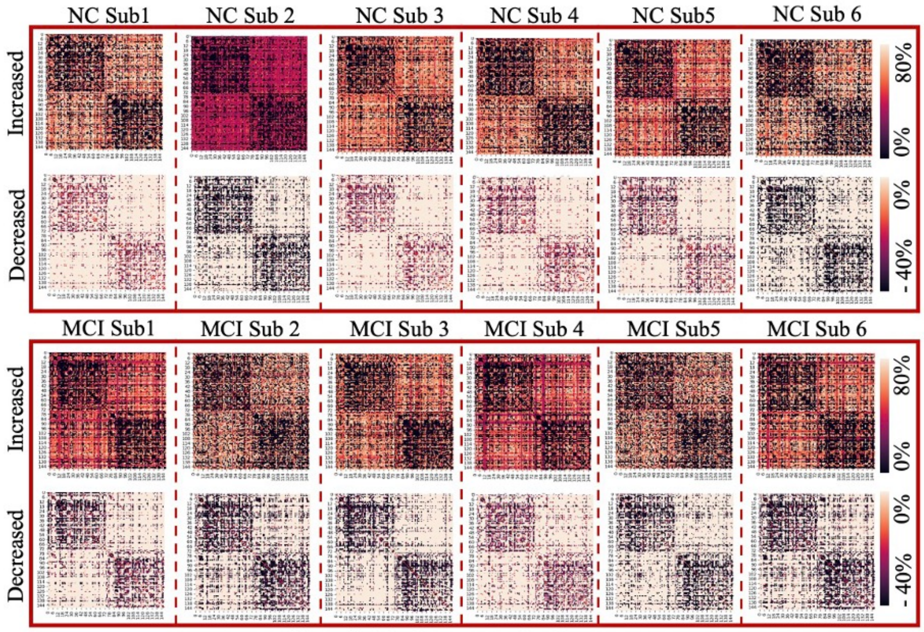
Study	Modality	Group			Method	Accuracy (%)			
		AD	MCI	NC		AD/ NC	MCI/ NC	AD/ MCI	pMCI/ sMCI
Dyrba et al. [108]	MRI+DTI+fMRI	28	-	25	ROI-based; Multi-kernel SVM	79.0	-	-	-
Cheng et al. [107]	MRI+PET+biomarkers	51	43^p+56^s	52	ROI-based; Doman transfer SVM	-	86.4	82.7	79.4
Yu et al. [114]	MRI+PET+biomarkers	50	97	52	ROI-based; Graph-guided learning	92.6	80.0	-	-
Shi et al. [115]	MRI + PET	51	43^p+56^s	52	ROI-based; Multimodal SDPN	97.1	87.2	-	78.9
Tong et al. [116]	MRI+PET+biomarkers+Genetics	37	75	35	Nonlinear Graph-fusion;	91.8	79.5	-	-
Aderghal et al. [112]	MRI+DTI	48	108	58	Hippocampus; CNN; Transfer learning	92.5	80	85	-
Liu et al. [86]	MRI+PET	93	76^p+128^s	100	Patch-based; Cascaded CNNs	93.3	83.0^p 64.0^s 74.3	-	-
Huang et al. [113]	MRI+PET	647	326^p+441^s	731	Hippocampus; CNN	90.1	87.5^p	-	76.9
Peng et al. [117]	MRI+PET+ Genetics	49	93	47	Structured sparsity; Kernel representation	96.1	80.3	76.9	-
Zheng et al. [56]	MRI+PET	-	51^p+75^s	-	Connectivity-based; Multi-task regression	-	-	-	79.4
Shao et al. [111]	MRI+PET	160	273^e+187^l	160	Hypergraph; Multi-kernel SVM	92.5	82.5	-	75.48
Proposed	MRI+DTI+fMRI	-	98	116	GBDM	-	**92.7**	-	-

Group: a^e+b^l: the number of **EMCI** is ***a*** and the number of **LMCI** is ***b***; c^p+d^s: the number of **pMCI** is ***c*** and the number of **sMCI** is ***d***.

Accuracy (%): a^p: the accuracy of CN/**pMCI** is ***a***; b^s: the accuracy of CN/**sMRI** is ***b***.



(a) Averaged Deep Brain Connectome

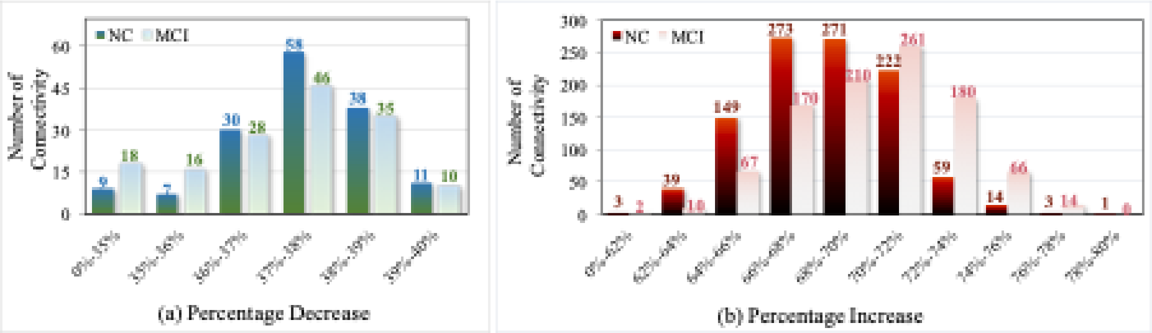


(b) Individual Deep Brain Connectome

Figure 3.5. The results of the learned Deep Brain Connectome and comparison with brain structural network. (a) shows the averaged results for NC group (the first row) and MCI group (the second row). For each group, the first column displays the group-wise structural network and the order of the brain regions listed in the matrix is the same as Destrieux atlas. The second column shows the learned deep brain connectome. Compared to brain structural network, the increased and decreased connectivity in deep brain connectome are shown in the third and fourth column. The top changed connectivity using the thresholds of 70% (for increased connectivity) and 37% (for decreased connectivity) are shown in the fifth and sixth column. (b) shows the increased and decreased connectivity of 6 randomly selected subjects from NC group (the first block) and MCI group (the second block).

3.3.5.1 Overall patterns of deep brain connectome

After applying the trained GBDM to each testing subject, the individual deep brain connectome - $\hat{A}$ can be computed via (3.6). The difference between $\hat{A}$ and the structural network reflects the influence of functional data in the deep structure-function fusion when classifying MCI from NC. Therefore, we are interested with the changed connectivity of $\hat{A}$ comparing to the corresponding structural network. Fig. 3.5 (a) shows the averaged results of different groups: NC and MCI. For each matrix, the order of the brain regions is the same as the order defined in Destrieux atlas [1]: the upper left/lower right represents the connectivity within the same hemisphere and the connections between different hemispheres are located in upper right/lower left. The first column displays the group-wise structural network and we can see that the overall patterns of two groups are similar: they show relatively stronger fiber connections in the same hemisphere [114]. The second column shows the obtained deep brain connectome. Through visual examination we can find that comparing to the structural network, the deep brain connectome is decreased within the same hemisphere and increased across different hemispheres. These interesting results suggest that after incorporating functional data to structural network, some crosshemisphere connectivity are strengthened to improve the classification power in deep brain connectome. To better demonstrate the inter- and intra- hemisphere patterns, we showed the increased and decreased connectivity separately at the third and the fourth columns in Fig. 3.5 (a). We can see that the increased connectivity is mostly composed of the ones connecting to different hemispheres (the third column), while the decreased connectivity is located within the same hemisphere (the fourth column). Note that in order to improve the robustness and exclude the potential noisy data, we only consider the increased/decreased connectivity which have changed by more than 5% on every subject in the testing dataset. Besides the averaged results, we also randomly selected 6 subjects from testing dataset for each group and showed the individual increased and decreased connectivity in Fig. 3.5 (b). The patterns of the altered connectivity of single subject are consistent with the averaged results. Moreover, we used medians of the changes (70% for increased and 37% for decreased) as the thresholds to select the top increased and decreased connectivity and showed them in the fifth and the sixth columns in Fig. 3.5 (a). Our results indicate that the number of top increased connectivity in MCI group is much more than that in NC group. To further analyze the group level differences, we examine the distribution of the changed connectivity in deep brain connectome based on the changing scale.



(a) Percentage Decrease

(b) Percentage Increase

Figure 3.6. Changing rate (%) for the increased connectivity (a) and decreased connectivity (b) of NC and MCI groups.

3.3.5.2 Changed connectivity in deep brain connectome

To further study the changed connectivity in deep brain connectome and better illustrate the group level differences, we calculated the distribution regarding the changing scale of increased and decreased connectivity (Fig. 3.6). For both NC and MCI groups, there are more increased connectivity ($\mathrm{NC} = 1034, \mathrm{MCI} = 980$ and

2014 in total) than decreased connectivity ($\text{NC} = 153, \text{MCI} = 153$ and 306 in total). The scale of changing for decreased connectivity mainly ranges from 35% to 40%, whereas increased connectivity has higher percentage of changing which is from 62% to 80%. The increased connectivity inclines to possess larger changing scale than the decreased ones. Another observation is that for increased connectivity, the changing scale of MCI group is higher than NC group.

3.3.5.3 Visualization of the brain regions involved in the top changed connectivity

For better visualizing the changed connectivity and the related brain regions, we adopt two strategies to project them back to brain space. We first show the top 10 increased and decreased connectivity with the largest changing scale for MCI and NC groups in Fig. 3.7 (a). It is consistent with Fig. 3.5 that the increased connectivity are mainly the ones connecting the regions on different hemispheres and the decreased connectivity are located within the same hemisphere. In addition, the top increased connectivity inclines to connect brain regions with long distance, while decreased ones mostly connect local areas.

The second way to project the changed connectivity is using the same threshold for the level of changing compared to the initial structural network. Here, to better demonstrate the differences between NC and MCI groups and the differences between increased and decreased connectivity, we used 72% (for increased connectivity) and 38% (for decreased connectivity) and the results are shown in Fig. 3.7 (b). For both MCI and NC groups, the number of increased connectivity is much more than the decreased ones. This implicates that to improve the classification performance, the learned deep connectome enhanced many structural connectivity by deep fusion of structural and functional data. Moreover, with the same threshold MCI group shows much more increased connectivity than NC group.

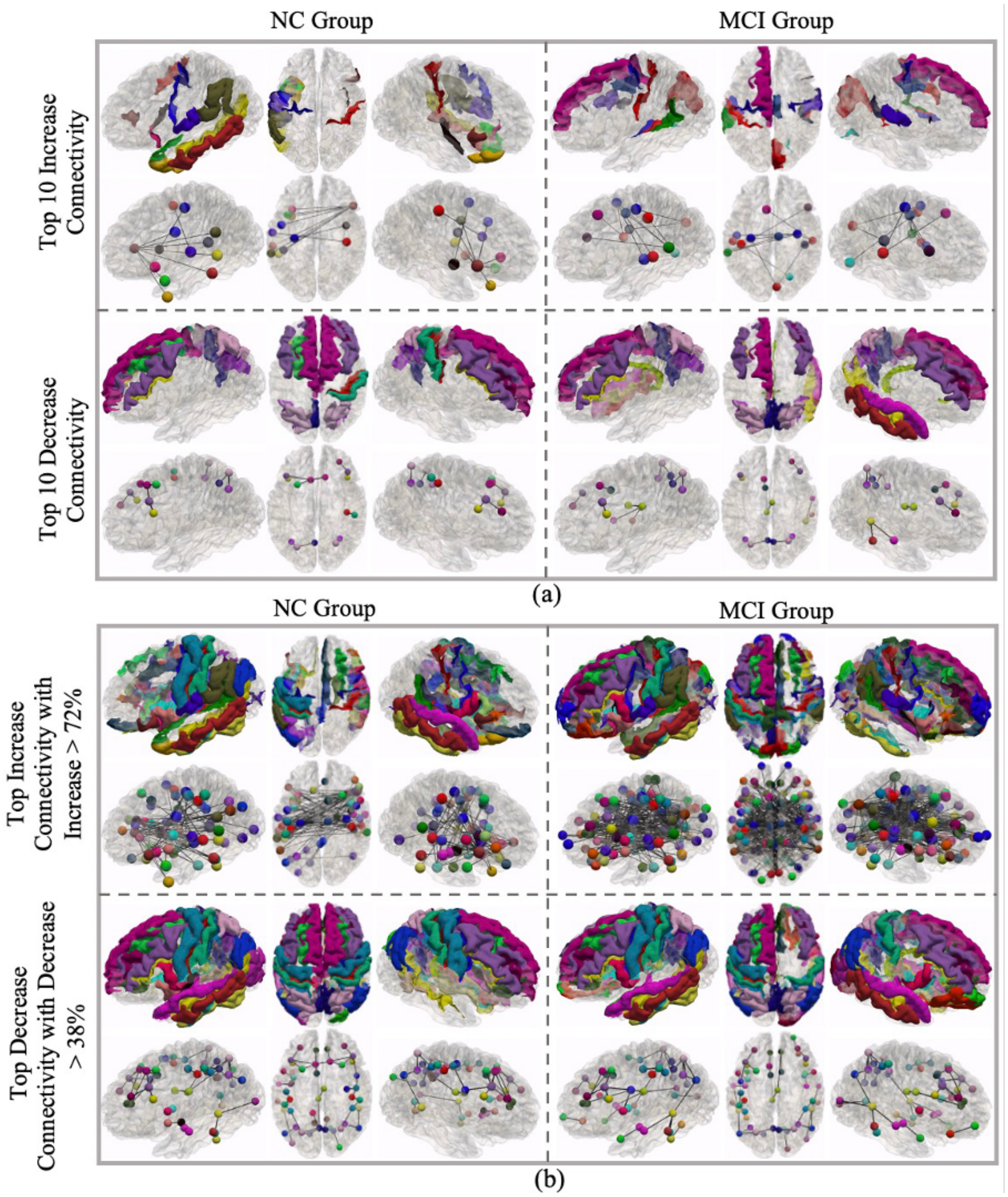

NC Group
MCI Group
Top 10 Increase Connectivity
Top 10 Decrease Connectivity
(a)
NC Group
MCI Group
Top Increase Connectivity with Increase > 72%
Top Decrease Connectivity with Decrease > 38%
(b)

Figure 3.7. The top increased/decreased connectivity and the involved brain regions for both MCI and NC groups. (a) shows the brain regions related to the top 10 increased and decreased connectivity and their connections. (b) shows the brain regions related to the increased and decreased connectivity with an averaged changing rate above 72% and 38%, respectively. For subfigures in both (a) and (b), the first row shows the involved regions and the second row highlights the regions as colored bubbles as well as the connections.

3.3.6 Model Evaluation

There are two key hyper-parameters in the proposed GBDM that may affect our results. The first is the σ in (3.3), which controls functional profile learning. The second parameter is the dimension of the output features at each graph convolutional layer, which directly affects graph convolution. In this paper, we used $G\left(F_1, F_2\right)$ to denote the architecture of graph convolutional network and F_l is the dimension of the output features at the lth convolutional layer. Since the dimension of the input data is 148 ×148, we will evaluate the influence using different feature dimensions $F_l=148 * a,(a=1,1.5,2, \ldots, 6)$. For each model setting we conducted 5 -fold cross-validation using the same input data. The influences of the two hyper-parameters on classification performance, overall patterns of the changed connectivity and the scale of changing are discussed in this section.

3.3.6.1 Influence on classification performance

Because the proposed GBDM is driven by the classification results, we firstly evaluate the influence of σ and feature dimensions on classification performance. We tried a spectrum of σ (from 1.0 to 3.0) with different feature dimensions (from $(148,296)$ to (444, 888)) and showed the classification performance in Table 3.2. In order to evaluate the classification performance in a more comprehensive way, we summarized accuracy (Acc), averaged precision (Pre), averaged recall (Rec), and averaged specificity (Spec) for each model setting. For accuracy, we showed the best, worst, and averaged results separately. The combination of $\sigma=2.0$ with G $(148,296)$ gives the best accuracy -92.7%. The other settings also showed decent performance in this work. In general, our proposed GBDM performs relatively stable within a wide range for the two key parameters.

Table 3.2. Classification Performance under Different Model Settings.

Model Setting		Performance (%)					
Feature dimension	σ	Acc[a]			Pre[b]	Rec[c]	Spec[d]
		Best	Worst	Ave[e]			
G(148, 296)	1.0	87.4	82.9	84.2	85.5	75.7	90.4
G(148, 296)	1.5	89.3	84.5	87.1	83.1	90.8	84.0
G(148, 296)	**2.0**	**92.7**	83.3	86.3	85.7	80.6	89.9
G(148, 296)	2.5	86.6	79.6	83.6	86.2	74.7	88.3
G(148, 296)	3.0	89.5	81.3	85.9	84.1	80.3	90.2
G(222, 444)	2.0	85.9	78.9	83.6	82.4	83.6	83.5
G(296, 592)	2.0	84.3	79.8	82.9	88.7	72.3	92.0
G(370, 740)	2.0	88.8	81.4	84.7	90.2	83.3	93.0
G(444, 888)	2.0	90.2	87.3	89.1	82.8	89.8	88.7

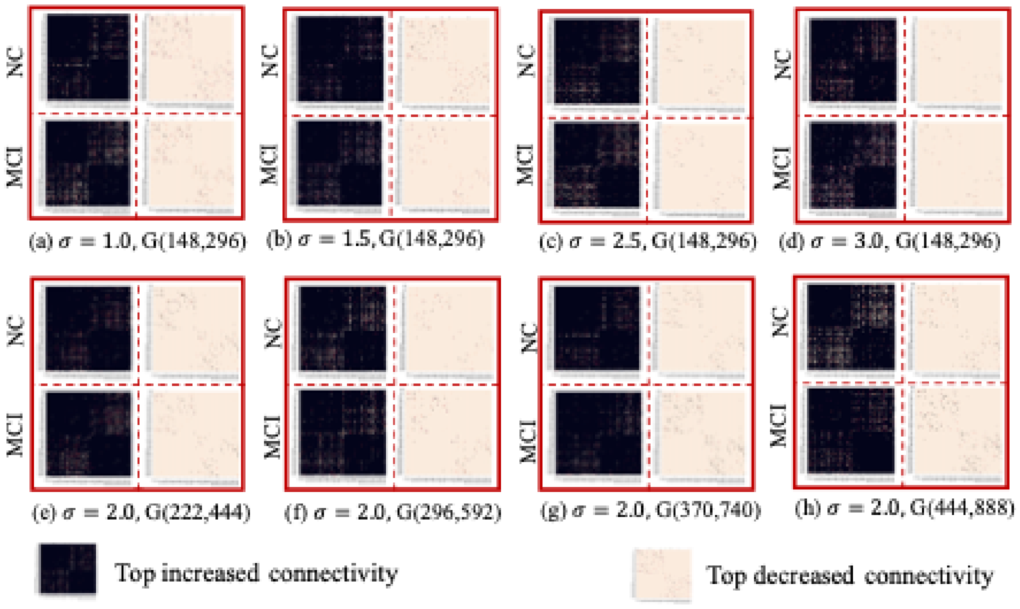
[a]Acc = Accuracy, [b]Pre = Precision, [c]Rec = Recall, [d]Spec = Specificity, [e]Ave = Average

3.3.6.2 Influence on the patterns of the changed connectivity

To demonstrate the influence of different parameters on the changed connectivity in deep brain connectome, we displayed the top increased and decreased connectivity obtained under different model settings in Fig. 3.8. In general, the patterns of the changed connectivity are very consistent: though the number and scale of the changed connectivity may be slightly different with different σ and feature dimensions, the top increased and decreased connectivity display the same inter- and intra-hemisphere patterns as Fig. 3.5.

3.3.6.3 Influence on the scale of the changed connectivity

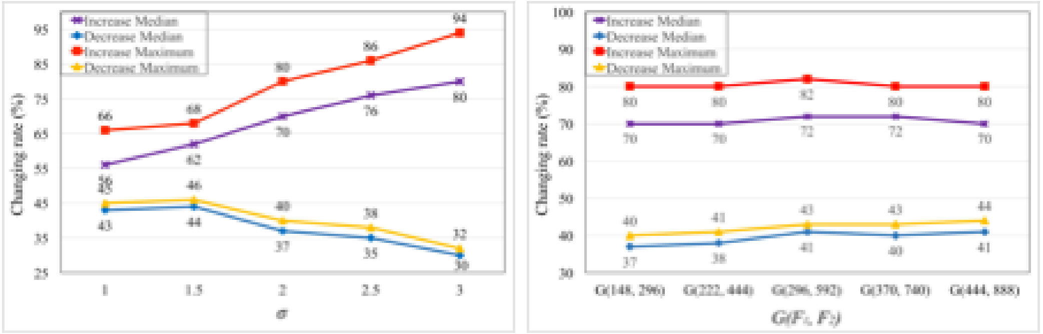
Here we will discuss the influence of σ and feature dimension on the scale of the changed connectivity. We projected the trends of maximum and median values of the changing scale in Fig. 3.9. Fig. 3.9 (a) shows the impact of σ with fixed feature dimension. It is obvious that as σ becomes larger, both the maximum and the median of the changing scale for increased connectivity become larger, while the decreased connectivity inclines to change less. One reason is that according to (3.3)



(a) σ = 1.0, G(148,296) (b) σ = 1.5, G(148,296) (c) σ = 2.5, G(148,296) (d) σ = 3.0, G(148,296)

(e) σ = 2.0, G(222,444) (f) σ = 2.0, G(296,592) (g) σ = 2.0, G(370,740) (h) σ = 2.0, G(444,888)

Figure 3.8. Top increased and decreased connectivity under different model settings. We used different combinations of σ and feature dimension to conduct our experiments.



(a) Different σ under *G(148,296)* architecture (b) Different feature dimensions with σ = 2.0

Figure 3.9. Influence of σ and feature dimension on changing scale of the altered connectivity. (a) shows the maximum and median of changing rate of the increased and decreased connectivity with a spectrum of σ (from 1.0 to 3.0) based on G(148, 296). (b) shows the maximum and median of changing scale of the increased and decreased connectivity of different feature dimensions with the same $\sigma = 2.0$.

the learned functional profile A^F is positive proportional to σ. Thus, a larger σ leads to an A^F with larger values. Because A^F contributes to the partial differences between structural connectivity and deep brain connectome - $\hat{A}$, the change of σ will affect the scale of the changed connectivity. Even so, more than 83% of the brain regions associated with the top 100 changed connectivity are still the same. Fig. 3.9 (b) shows the impact of feature dimension with fixed σ. The maximum and median of the changing scale for both increased and decreased connectivity are similar across different model settings. These results suggest that the dimension of the output features seems less sensitive than the other hyper-parameter $-\sigma$.

3.4 Discussion

The core component of GBDM is the deep fusion of functional data and structural network: driven by the classification result, the graph that represents the learned connectome is iteratively updated based on functional features upon current network topology. Therefore, an interesting question would be how to understand and interpret the newly learned brain network - deep brain connectome. In the past decades, numerous connectome-based studies have reported and confirmed both structural and functional alterations in AD/MCI patients [67, 52]. However, the learned deep brain connectome cannot be treated as a simple combination of structural and functional connectivity. In this work, we conducted convolutional operations (see (3.8)) on graph-based neighbors which are defined by the adjacency matrix - $\hat{A}$. Essentially, the entry in $\hat{A}$ is proportional to the output at each layer and a larger connectivity in deep brain connectome indicates this connectivity may be potentially important in the classification task. The most brain regions related to the top increased connectivity are widely known for their close relations to AD/MCI development, such as orbital part of the inferior frontal gyrus (G_front_infOrbital) [115] and temporal regions. This result suggests the training process of GBDM tends to alter brain network topology (initialized using structural network) to provide more efficient paths for convolution operations: by increasing or decreasing connectivity in current brain network

($\hat{A}$), the way of integrating information from neighbors can be optimized to improve the overall classification performance. In addition, it is not necessarily true that the structural and functional connectivity are always consistent as functional connectivity can be observed between brain regions with no direct connections [66, 19, 68]. Therefore, when incorporating functional data, the abovementioned training process will increase the connectivity between the regions in different hemispheres that have little or no structural connectivity (Fig. 3.5). Another interesting observation is the top increased connectivity inclines to connect brain areas with long distance, while decreased ones mostly connect local regions. In our graph convolutional operations, increasing/decreasing a connectivity will give more/less consideration to the related regions when combining functional features. Our results in Fig. 3.5 and Fig. 3.7 indicate the deep brain connectome tries to enhance the weight of long-distance connections instead of short-distance connections. Recent studies show that long-distance connections play a critical role in functional diversity and complexity of dynamics. Especially for AD and MCI, the severity of disease and cognitive impairment was more associated to long-distance connected regions [116]. In general, the derived deep brain connectome reflects which connections/paths that are“important” to the classification task.

The core idea of deep brain connectome is to identify the disease related brain network by learning the connectome topology (using both structural and functional information) instead of fixing the predefined brain network. In this work, our proposed model has been applied to MCI/NC classification task, however, it can be easily extended to other tasks. In many brain disorders, pathological changes tend to be “global” instead of “local”: the structural and functional alterations are not equally distributed over the brain; instead, they often spread via structural or even functional network to influence other brain regions. Thus, our deep brain connectome can be a promising approach to explore the underlying relations between brain structural and functional perturbations at network level in both neurological and psychiatric diseases. In addition, by using appropriate label information including clinical status, subtyping or other cognitive/behavior measures, deep brain connectome can be used in the study of brain development, aging, disease progression and many other applications. For example, classification loss is used to train the graph topology, but other loss options can be adopted based on different tasks, such as regression loss (e.g., for continuous measures) and task-specific loss (e.g., a combined loss for multiple tasks).

3.5 Conclusion

It is widely believed that the AD/MCI related brain alterations involve both brain structure and function. However, how to computationally model the complex and potentially non-linear relations between structural and functional data and integrate them at network level is still challenging. Inspired by the most recent development of deep learning approaches, in this work we developed a graph-based deep model - GBDM to classify MCI patients from normal controls. We constructed a graph convolutional network with trainable topology which is learned from functional features and structural network simultaneously. Comparing to the initialized structural network, the learned new brain network - deep brain connectome shows increased connections connecting to different hemispheres and regions with long-distance. Our developed GBDM achieves 92.7% classification accuracy on ADNI dataset and outperforms most recent studies. Given the complementary information stored in multimodal data, we envision that our proposed deep brain connectome can provide a promising way for the deep fusion of brain structure and function.

CHAPTER 4

DISEASE2VEC: ENCODING ALZHEIMER'S PROGRESSION VIA DISEASE EMBEDDING TREE

Another significant challenge in the study of brain diseases, particularly neurodegenerative disorders, such as Alzheimer's disease (AD), arises from the extended preclinical phase, which spans from clinical asymptomatic stages to severe impairment. Although a variety of predictive approaches have been proposed and evaluated in terms of their prediction capability for AD and its precursor – mild cognitive impairment (MCI). Most of them focused on prediction or identification of statistical differences among different clinical groups or phases, especially in the context of binary or multi-class classification. The continuous nature of AD development and transition states between successive AD related stages have been typically overlooked. Though a few progression models of AD have been studied recently, they were mainly designed to determine and compare the order of specific biomarkers. How to effectively predict the individual patient's status within a wide spectrum of continuous AD progression has been largely understudied. To solve this problem, in **Chapter 4**,we developed a novel learning-based embedding framework to encode the intrinsic relations among AD related clinical stages by a set of meaningful embedding vectors in the latent space (Disease2Vec). We named this process as disease embedding. By Disease2Vec, our framework generates a disease embedding tree (DETree) which effectively represents different clinical stages as a tree trajectory reflecting AD progression and thus can be used to predict clinical status by projecting individuals onto this continuous trajectory. Through this model, DETree can not only perform efficient and accurate prediction for patients at any stages of AD development, but also provide richer status information by examining the projecting locations within a wide and continuous AD progression process. Moreover, the proposed framework is a general one that can be applied to various diseases.

4.1 Introduction

Alzheimer's disease (AD) is the most common cause of dementia that cannot be prevented, cured, or even slowed. Earlier studies have shown that AD pathogenesis involve widespread alterations in brain structure and/or function, such as hippocampi [117], gray matter atrophy [118], white matter disruption [119] and abnormal functional connectivity in default mode network (DMN) [120]. Based on these brain alterations, many approaches have been developed for early diagnosis of AD and its prodromal stage – mild cognitive impairment (MCI), such as voxel-based analysis [121], tract-based spatial statistics [122], and recently developed machine learning/deep learning-based models [123, 41, 105]. However, as a neurodegenerative disorder with a long pre-clinical period, the spectrum of AD spans from clinically asymptomatic to severely impaired [124]. For example, heterogeneity in clinical presentation, rate of atrophy and cognitive decline [125] may occur in the prodromal stage of AD [126]. Furthermore, individual variations may also contribute to the heterogeneity of AD: earlier studies suggested that the gap between cognitive function and brain pathology (i.e., cognitive reserve) is typically larger in highly educated individuals [127]. In general, traditional predictive approaches (e.g., classification-based models) may be limited in describing the continuum of AD development and individual variations in clinical prediction. To address this potential limitation, hypothetical models [128] for AD progression have been proposed and followed by various progression studies using cross-sectional or short-term follow-up dataset. These attempts include regression-based models [129], event-based models [130] and other computational models [131]. Nevertheless, most of them were designed to determine the order of biomarkers. Because these models consider different measures or biomarkers separately, they create a different trajectory for each biomarker. Consequently, different models and assumptions may lead to inconsistent results and interpretations [130]. More importantly, previous AD progression models are based on population analysis, they cannot be directly used for individualized diagnosis and prediction.

Recent advancements in deep learning have brought about a paradigm shift in representation learning, leading to the emergence of powerful embedding techniques across various domains. Notably, word embedding methods have made remarkable contributions in the field of natural language processing (NLP) [132]. These methods aim to generate dense, continuous vector representations of words, effectively representing and capturing the complex semantic relationships among them. That is, the words sharing similar semantic meanings can be projected as the vector representations that are closely situated in a high-dimensional space. These embedding methods have demonstrated exceptional performance across a range of downstream tasks [132]. Similarly, a groundbreaking approach known as cortex2vector [133] has recently been introduced. This method focuses on encoding the cortical folding patterns into a group of anatomically meaningful embedding vectors. By leveraging this technique, the anatomical similarity of different brain landmarks can be effectively represented by the similarity of their corresponding embedding vectors. The core idea underlying these embedding methods lies in the direct shaping of the latent space (embedding space), aligning it with the semantic space. By mapping complex data into the latent space, different positions within this space correspond to distinct semantic or anatomical information. This alignment enables efficient representation and analysis

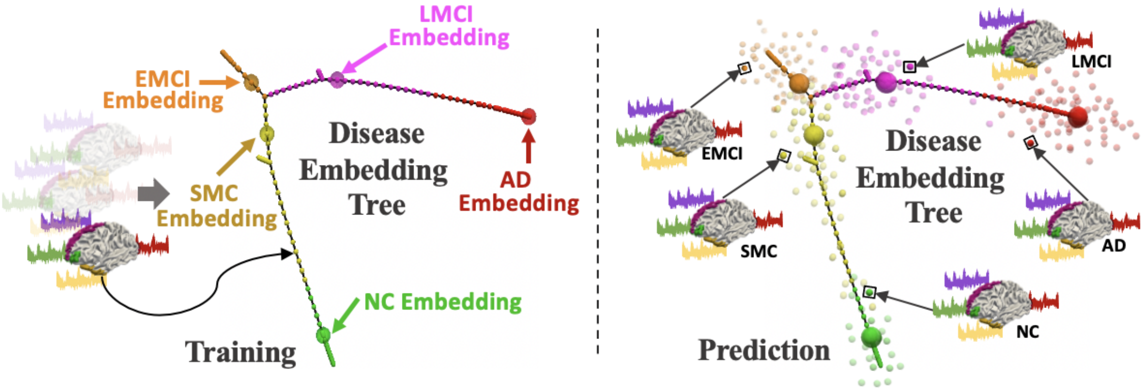


Figure 4.1. Training: we used functional connectivity as input and learned a Disease Embedding Tree (DETree) to model the entire progression of AD in the latent space. In the tree structure, each small bubble represents a single subject, and the colors indicate different clinical groups, including normal control - NC (green), significant memory concern - SMC (yellow), early MCI - EMCI (orange), late MCI - LMCI (pink) and AD (red). Each edge in the DETree indicates that the connecting two nodes have higher similarity in the latent space. The five larger bubbles represent the learned group embeddings. Prediction: During the prediction, new patients will be projected into the latent space which are represented as scattered bubbles. The color of the bubble indicates the true label, the location of the bubble shows its state in the entire development process from NC to AD, and the prediction of the bubble is based on the nearest group embedding.

of complex relationships and therefore enhances our ability to explore and understand intricate data structures.

Inspired by the abovementioned remarkable successes of embedding methods, in this work, we designed a new learning-based embedding framework to encode the entire AD progression by a set of meaningful embedding vectors in the latent space (Disease2Vec). By employing this approach, the latent space is aligned with the AD progression trajectory, allowing for the representation of intrinsic relationships between different clinical stages through the learned embedding vectors. Fig. 4.1 provides a visual depiction of the core idea behind our framework. During the training process, we introduce a novel ordered embedding method that shapes the latent space in alignment with the disease development process. This results in a set of learnable embeddings for different clinical groups. These group embedding vectors are orderly arranged in the latent space, directly corresponding to different stages of disease progression. To effectively capture individuality, we transform the input features of individuals into the latent space to obtain individual embeddings. Through the learning process, the distribution of these individual embeddings in the space reflects the corresponding individual's clinical status within the entire disease development process. The proposed framework jointly learns the clinical group embeddings and individual embeddings to better shape the latent space. As a result, we obtain a tree-based trajectory within the latent space, known as the Disease Embedding Tree (DETree). This trajectory effectively integrates AD progression modeling and individual prediction. During the prediction process, new individuals can be projected onto the continuous trajectory of the learned DETree. This allows our model not only to assign clinical groups to new patients but also to indicate their clinical status throughout the entire development process, spanning from normal cognition (NC) to AD. With the learned DETree, our model achieves a relatively high classification accuracy – 77.8% for multi-class classification (NC vs. SMC vs. EMCI vs. LMCI vs. AD), compared to other established machine learning/deep learning methods [105, 127, 134, 135, 136, 137, 138, 139, 140, 141, 142]. Furthermore, the proposed DETree framework is versatile and can be adapted to a wide range of neurodevelopmental, neurodegenerative, and psychiatric disorders that exhibit multiple clinical stages during the development.

4.2 Materials and Methods

4.2.1 Data

4.2.1.1 Datasets Description and Data Pre-processing

In this work, we used 266 subjects (60NC, 34SMC, 51EMCI, 62LMCI and 59AD) from the ADNI dataset (http://adni.loni.usc.edu/). Each subject has both structure MRI (T1-weighted) and resting state fMRI (rs-fMRI) data. For T1-weighted MRI, FOV $= 240 \times 256 \times 208$ mm^3, voxel size = 1.0 mm isotropic, and TR = 2.3 s. The rs- fMRI data has 197 volumes, FOV $= 220 \times 220 \times 163$ mm^3, voxel size = 3.3 mm isotropic, TR = 3 s, TE = 30 ms and flip angle = 90°. We followed the standardized preprocessing procedures adopted in [143, 144] for imaging data. Specifically, we applied skull removal for both T1 and rs-fMRI modalities. And for rs-fMRI images, the first 6 volumes were discarded during preprocessing procedures to ensure magnetization equilibrium. Then we applied spatial smoothing, slice time correction, temporal pre-whitening, global drift removal and band pass filtering (0.01-0.1 Hz). All these preprocessing steps are implemented using FMRIB Software Library (FSL) (https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/) FEAT. For T1 images, we conducted segmentation by FreeSurfer package (https://surfer.nmr.mgh.harvard.edu/). After the segmentation, we adopted the Destrieux Atlas for ROI labeling, and the brain cortex is partitioned into 148 regions.

4.2.1.2 Generation of Functional Connectivity

We calculated averaged fMRI signal for each brain region. Previous studies [145, 56] suggested that for rs-fMRI 14 time points (when TR = 2 s) are sufficient to capture functional dynamic patterns. To enlarge the dataset, we divided the signal into four non-overlapping segments and each segment has 45 time points. We used

Pearson Correlation Coefficient to calculate functional connectivity for each of the four groups of the signal segments and obtained four functional connectivity matrices for each subject. These functional connectivity matrices were vectorized and used as input of our model.

4.2.2 Method Overview

We proposed a DETree framework to represent the continuum of AD development process as a tree structure embedded in a latent space. Here, an embedding is an abstract representation defined in latent space that is associated with a specific clinical stage (group embedding) or a specific individual (individual embedding). We parameterized a set of group embeddings as hidden variables in latent space (Section 4.2.3) and used the order information of clinical groups (NC → SMC → EMCI → LMCI → AD) to guide the embedding process (Section 4.2.4). In general, the proposed model aims to learn a deep representation of the input signals in a latent space that is specially optimized for both tasks simultaneously: the individual prediction and the AD progression learning. As a result, on the learned tree structure, the patients with similar clinical status are close and distant otherwise. Moreover, DETree can predict the clinical stage for a new patient by projecting it to the appropriate location on the learned tree structure (Section 4.2.4). Next, we will present the details of DETree and its predictive capability for new patients.

4.2.3 Disease Embedding Learning

Let $\{(x_i, y_i)\}_{i=1}^{n}$ be the training data consisting of n labeled data with the ith input $x_i \in \mathcal{R}^d$ and class label $y_i \in \{1, \ldots, C\}$ with C disease stages. To maintain representative instances for different disease stages, we parameterized and learned a set of embeddings in the latent space and used embedding matching for classification.

First, we learned a non-linear function $h(x, \theta) : \mathcal{R}^d \rightarrow \mathcal{R}^k$ to transform any given input $x \in \mathcal{R}^d$ to a latent space $\mathcal{R}^k$ with learnable model parameter θ to obtain individual embedding. And then, we defined a set of clinical group embeddings as $\mathcal{E} = \left\{e_{i,j} \in \mathcal{R}^k \mid i = 1, 2, \cdots, C; j = 1, 2, \cdots, K\right\}$ where K is the number of the embeddings in each class. With the help of the non-linear transformation function $h(x, \theta)$ and the set of embeddings $\mathcal{E}$ in the latent space, we can make prediction for any given data. Specifically, given an input data $x \in \mathcal{R}^d$, we first generated its representation (individual embedding) $h(x, \theta)$ in latent space, then we compared the individual embedding with all clinical group embeddings and classified it to the category y, which is the nearest clinical group that the embedding belongs to:

$$y = \operatorname{argmin}_{i \in \{1,2,\cdots C\}} \min_{j \in \{1,2,\cdots K\}} \left\| h(x, \theta) - e_{i,j} \right\|_2^2 \quad (4.1)$$

The network parameters θ and clinical group embeddings $\mathcal{E}$ can be trained jointly in an end-to-end manner, which can make the model $h(x, \theta)$ and clinical group embeddings interact with each other for better performance. To train the model, we need to define a proper loss function such that 1) it is differentiable with respect to θ and $\mathcal{E}$, and 2) it should be closely related to the classification accuracy.

Embedding Learning Based Cross Entropy Loss. In our DETree model, we used distance to measure the similarity between the individual embeddings and the clinical group embeddings. The class label of the clinical group embedding $e_{i,j}$ can be denoted by $y_{i,j}$, to indicate the jth embedding of class y_i. Thus, the probability of an input x belongs to class y_i (i.e., $e_{i,j}$ is the nearest embedding of x) is formulated as:

$$P\left(y_{i,j} \mid x\right) = \frac{\exp\left\{-\alpha \left\| h(x, \theta) - e_{i,j} \right\|_2^2\right\}}{\sum_{l=1}^{C} \sum_{m=1}^{K} \exp\left\{-\alpha \left\| h(x, \theta) - e_{l,m} \right\|_2^2\right\}} \quad (4.2)$$

where α is a hyper-parameter that controls the hardness of distance in probability assignment. Given the definition of $P\left(y_{i,j} \mid x\right)$, we can further define the probability of an input x belonging to the category $c \in \{1, 2, \cdots, C\}$ as:

$$P(c \mid x) = \sum_{j=1}^{K} P\left(y_{c,j} \mid x\right) \text{(4.3)}$$

Then, we defined a classification loss function based on the probability $P(c \mid x)$ and named it as embedding learning based cross entropy loss given by:

$$\mathcal{L}_{\mathcal{E}}((x, y); \theta; \varepsilon) = -\frac{1}{C} \sum_{c=1}^{C} I(c = y) \log P(c \mid x) \text{(4.4)}$$

where indicator function $I(c = y)$ is 1 if predicator $c = y$ is true and 0 otherwise.

From (4.2), (4.3) and (4.4), we can see that optimizing the embedding based cross entropy loss essentially corresponds to decreasing the distance between the individual embedding $h(x, \theta)$ of input sample x and the clinical group embedding vector, which comes from the true category of x. By this way, the distance of two input samples at the same disease stage will be small in the latent space, and the disease related representative clinical group embeddings can be automatically learned from data.

To improve the generalization performance and prevent over-fitting, we also proposed a new embedding-based regularization term:

$$\mathcal{L}_{\mathcal{ER}}((x, y); \theta; \varepsilon) = \|h(x, \theta) - e_{y,*}\|_2^2 \text{(4.5)}$$

where $e_{y,*}$ is the closest group embedding of $h(x, \theta)$ with class label y. The regularization term pulls the individual embedding $h(x, \theta)$ of input sample x close to its corresponding clinical group embedding, making the individual embeddings within the same class more compact, so it is beneficial for classification.

4.2.4 Ordered Embedding Constraint

The class labels y provides not only the separability of their inputs, but also the underlying relationship of the clinical groups, which corresponds to different disease stages during the progression of AD. It is generally assumed that the ordering of the clinical groups is NC → SMC → EMCI → LMCI → AD. Even though the ordering of each input sample is unknown, the ordering of the classes can still provide valuable information to guide the embedding learning. To take advantage of this prior knowledge, we constructed an affinity matrix $\mathcal{A} = \left[a_{(i,j),(i',j')}\right] \in \mathcal{R}^{N\times N}$ for the similarity among embedding class labels. $N = C \times K$ is the total number of embeddings, where C is the number of clinical stages and K is the number of the embeddings in each class. $a_{(i,j),(i',j')} = 1$ if the $(i,j)^{\text{th}}$ embedding and the $(i',j')^{th}$ embedding are from the same class, that is $y_i = y_{i'}$, $a_{(i,j),(i',j')} = 0.5$ if y_i is the neighbor of $y_{i'}$ in the ordering of class labels, and 0 otherwise.

To leverage this prior information for learning the path of AD progression, we added an additional neural network layer with softmax function onto the embeddings to link the clinical group embeddings and the different classes (stages) of AD. As a result, the output probability of clinical group embedding $e_{i,j}$ belonging to the class c is formulated as:

$$O_c\left(e_{i,j}; W, b\right) = \frac{\exp\left\{\left(W_c^T e_{i,j} + b_c\right)\right\}}{\sum_{l=1}^{C} \exp\left\{\left(w_l^T e_{i,j} + b_l\right)\right\}} (4.6)$$

where $\{W_l, b_l\}$ are the parameters of the neural network layer. The final prediction is:

$$y_{i,j} = \underset{c\in\{1,2,\cdots c\}}{\operatorname{argmax}} O_c\left(e_{i,j}; W, b\right) (4.7)$$

According to (4.6) and (4.7), the classification loss of clinical group embeddings is defined as:

$$\mathcal{L}_{\mathcal{O}}\left(e_{i,j};W,b\right)=-\frac{1}{c}\sum_{c=1}^{C}I(c=i)\log O_{c}\left(e_{i,j};W,b\right)\tag{4.8}$$

Then, we proposed the following regularization term to incorporate the ordering information of the class labels in terms of the affinity matrix $\mathcal{A}$ based on the manifold assumption: if two labels are similar, their probabilities of predictions should be close. The regularization term is then formulated as:

$$\mathcal{L}_{\mathcal{OR}}(\mathcal{E};W,b)=\operatorname{trace}\left(OL_{y}O^{T}\right)\tag{4.9}$$

where $O=[O_1;O_2;\cdots;O_C]\in\mathcal{R}^{C\times(C\times K)}$ and $O_c=\left[O_c\left(e_{i,j};W,b\right)\right]_{\{i,j\}}\in\mathcal{R}^{1\times(C\times K)},\forall c$, $L_y=\mathcal{D}-\mathcal{A}$ is the graph Laplacian matrix of $\mathcal{A}$ and $\mathcal{D}=\operatorname{diag}\left(\sum_{j=1}^{n}\mathcal{A}_{i,j}\right)$.

Together with (4.4), (4.5), (4.8), (4.9) in hand, we are now ready to formulate our DETree model with the loss function defined as:

$$\mathcal{L}=\sum_{i=1}^{n}\left[\mathcal{L}_{\mathcal{E}}\left(\left(x_i,y_i\right);\theta;\mathcal{E}\right)+\beta\mathcal{L}_{\mathcal{ER}}\left(\left(x_i,y_i\right);\theta;\varepsilon\right)\right]+\gamma\sum_{i=1}^{C}\sum_{j=1}^{K}\mathcal{L}_{\mathcal{O}}\left(e_{i,j};W,b\right)+\delta\mathcal{L}_{\mathcal{OR}}(\mathcal{E};W,b)\tag{4.10}$$

This loss function (4.10) is derivable with respect to $\theta,\mathcal{E},W$ and b. The whole model can be trained in an end-to-end manner. Once the model is trained, the latent space will demonstrate a clear alignment with the different stages of AD progression. The clinical group embeddings will be organized in a meaningful sequence within the latent space, and the distances between individual embeddings will accurately capture the relatedness of disease states among corresponding individuals throughout the entire course of the disease. To visually represent the intricate relationships among individuals and offer a clear depiction of each individual's position throughout the disease progression, we attempt to create a structured tree-like representation based on these learned embeddings. In this tree structure, individuals that share similar disease states are connected by edges, providing a visual reflection of the disease's developmental process. To achieve this, we initiated the process by computing pairwise distances between individual embeddings, resulting in an $n \times n$ distance matrix, where n is the number of embeddings. Subsequently, upon the obtained distance matrix we generated a minimum spanning tree using Kruskal's algorithm. Within the tree structure, the connecting individuals have shortest latent distances and share similar disease status. This approach effectively provides a visual and structural representation of the disease progression, facilitating a deeper understanding of the relationships among individuals at different stages of the disease and their positions within the progression. We named it Disease Embedding Tree (DETree).

For a new patient x, the DETree can provide two sets of predictions. Firstly, we can obtain the probabilities of assigning the new patient x to each of the given clinical groups using (4.3). Based on these probabilities, we can make the best prediction regarding the clinical group that patient x belongs to. Secondly, DETree enables us to determine the location of the individual patient within the learned tree using the function $h(x, \theta)$. This location reflects the specific stage in the progression of AD where the patient is situated. By utilizing these predictions, we can gain valuable insights into both the patient's clinical group assignment and their disease progression stage.

4.3 Results

4.3.1 Experimental Setting

Data Setting. In this work, we used 266 subjects (60 NC, 34 SMC, 51 EMCI, 62 LMCI, 59 AD) in our experiment. Based on Section 4.2.1.2 each subject has four functional matrices and we obtained 1064 data samples in total. In our experiments, the training, validation, and testing datasets were split according to subjects, that is, four matrices of the same subject will be divided into the same dataset. As the functional matrix is symmetric, to reduce the redundant data, we used the vectorized upper triangle of each matrix as input features.

Model Setting. In this work, the non-linear function $h(x, \theta)$ was implemented by 6-layer fully connected network. The dimensions of the fully connected network are 1024-512-256-64-16- k, where k is the dimension of the latent space (Section 4.2.3). We tested $k = 5, 10, 15, 20, 25$. We showed the results of $k = 25$ which gives the best classification performance in Section 4.3.2 and Section 4.3.3, and compared the results of $k =$ 5, 10, 15, 20 and 25 in Section 4.3.4. Activation function Relu and Batchnorm were used at each layer. C = 5 is the number of classes (NC/SMC/EMCI/LMCI/AD). For each class, we initialized one learnable group embedding ($K = 1$). We conducted hyper-parameter tuning for α, β, γ, and δ by searching a grid of powers of 10 within the range of 10^{-4} to 10^{1}. The selection of the best hyper-parameter values was based on the performance of models using the training and validation datasets, resulting in $\alpha = 1.0, \beta = 0.001, \gamma = 1.0$ and $\delta = 1.0$. The entire model was trained in an end-to-end manner. Adam optimizer was used to train the whole model with standard learning rate 0.001 , weight decay 0.01 , and momentum rates $(0.9, 0.999)$.

4.3.2 Classification Performance

In this section, we showed the classification performance of the proposed DE-Tree. For fair comparisons, we used two strategies to compare the proposed method with other widely used methods. Firstly, we repeated experiments 5 times with random seeds to compare the results with other four broadly used machine learning methods including support vector machine (SVM), k-nearest neighbors (KNN), logistic regression and random forest. We conducted grid searching using training and

Table 4.1. Classification Performance of DETree and Four Traditional Machine Learning Methods. The orange shade and blue shade highlight the best and the second-best results, respectively.

Method	**F1**						**Acc (All)**
	All	**AD**	**LMCI**	**EMCI**	**SMC**	**CN**	
SVM	0.658±0.05	0.684±0.06	0.641±0.05	0.630±0.05	0.659±0.05	0.675±0.04	0.672±0.03
KNN	0.526±0.05	0.427±0.09	0.576±0.05	0.546±0.07	0.547±0.08	0.535±0.04	0.532±0.04
Logistic Regression	0.642±0.03	0.651±0.06	0.634±0.04	0.627±0.05	0.649±0.05	0.651±0.03	0.649±0.02
Random Forest	0.422±0.03	0.377±0.05	0.516±0.07	0.426±0.04	0.436±0.06	0.354±0.04	0.428±0.03
DETree (segmented)	**0.777**±0.01	**0.785**±0.03	**0.762**±0.03	**0.801**±0.06	**0.765**±0.02	**0.773**±0.03	**0.778**±0.02

validation datasets to select the best model parameters for the four models. As a result, the SVM with linear kernel, regularization strength = 0.8, the KNN with $k = 9$, the logistic regression with lbfgs solver, 12 regularization, regularization strength = 1.2, and random forest with 100 trees obtained the optimal performance. The classification performance was measured by F_1 scores: $F_1 = 2\times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$ and accuracy (Acc). The results are showed in Table 4.1. We can see that the F_1 score of DETree model is over 0.75 which is more than 10% higher than the second-best results. And for some classes it can reach 0.80 , which is outstanding in multi-class classification of AD and significantly outperforms the other four methods.

Secondly, we compared the multi-class classification performance with latest deep learning methods on AD and reported the results in Table 4.2. As shown in Table 4.2, [20] obtains a very high F_1 score for AD group, however the F_1 scores for other groups are considerably lower. Although the total accuracy of [30] is slightly higher than our results ($0.780 > 0.778$), it is important to note that they only considered three classes, whereas our approach encompasses five classes in this study. In comparison with these methods, our proposed approach not only achieves a high

Table 4.2. Classification Performance of DETree and Other Deep Learning Methods. cMCI/pMCI: MCI patient who converted to AD within 36 months; sMCI: MCI patients who didn't convert to AD within 36 months. EMCI/LMCI: early/late MCI. The orange shade and blue shade highlight the best and the second-best results, respectively.

Study	**Modality**	**Participants (n)**	**Method**	**Task**
				Performance
Amorosoa et al. (2018) [134]	Predefined Features	60AD/60NC/ 60cMCI/60MCI	Deep Random Forest	AD vs. cMCI vs. sMCI vs. NC F_1: 0.805/0.518/0.305/0.525
Zhou et al. (2018) [146]	MRI, PET, SNP	190AD/226NC/ 157pMCI/205sMCI	Multi-modal Fusion	AD vs. pMCI vs. sMCI vs. NC Acc: 0.574/0.622/0.342/0.625
Brand et al. (2019) [136]	MRI, SNP	412(AD+MCI+NC)	Joint Regression-Classification	AD vs. MCI vs. NC F_1: 0.566/0.513/0.683
Lei et al. (2020) [127]	MRI	192AD/402MCI/ 220NC	Multiple Templates, Adaptive Feature Selection	AD vs. MCI vs. NC Acc: 0.775
Wang et al. (2020) [123]	rs-fMRI	253NC/45EMCI/ 88LMCI	Deep Autoencoder	LMCI vs. EMCI vs. NC Acc: 0.730
Puente-Castro et al. (2020) [137]	MRI	297AD/921MCI/ 525NC	Multi-Plane Features, Transfer Learning	AD vs. MCI vs. NC F_1: 0.433/0.760/0.616
Liu et al. (2021) [138]	MRI	90AD/136MCI/ 266NC	Depthwise Separable Convolution (kernel size: 3 × 3; maximum pooling size: 2 × 2)	AD vs. MCI vs. NC Acc: 0.780
Xu et al. (2021) [139]	MRI	85AD/244MCI/ 133NC	Tresnet, Selective Kernel module (kernel size: 3 × 3 and 1 × 1)	AD vs. MCI vs. NC Acc: 0.632
Lin et al. (2021) [147]	MRI, PET, CSF, genetic data	105AD/441MCI/ 200NC	Linear Discriminant Analysis, PCA, Multimodal Fusion	AD vs. MCI vs. NC Acc: 0.667
Lin et al. (2021) [147]	MRI, PET, CSF, genetic data	105AD/200NC/ 110pMCI/208sMCI	Linear Discriminant Analysis, PCA, Multimodal Fusion	AD vs. pMCI vs. sMCI vs. NC Acc: 0.573
Mulyadi et al. (2023) [141]	MRI	1540 (AD+MCI+NC)	Clinically Guided Prototype Learning	AD vs. MCI vs. NC Acc: 0.632
Sudharsan et al. (2023) [142]	MRI	80AD/84MCI/80NC	Regularized Extreme Learning Machine	AD vs. MCI vs. NC Acc: 0.628
Proposed	rs-fMRI	59AD/62LMCI/ 51EMCL/34SMC/ 60NC	Disease Embedding, DETree	AD vs. LMCI vs. EMCI vs. SMC vs. NC F_1: 0.785/0.762/0.801/0.765/ 0.773 Acc: 0.778

overall accuracy but also maintains high accuracies for each class without significant disparities.

4.3.3 The Learned Disease Embedding Tree

In addition to its outstanding classification performance, our DETree's most significant contribution lies in the introduction of a novel ordered embedding method to direct shape the latent space (embedding space). Through this method, the latent space is effectively aligned with the AD progression trajectory and a tree structure is learned to model the entire spectrum of AD progression. To evaluate the effectiveness of this method, we analyzed the results from two different perspectives.

Firstly, we conducted analyses from a group-level perspective. The well-trained model mapped the samples at different disease stages into the embedding space. We attempted to evaluate the alignment between embedding space with the AD progression by visualizing the feature distribution of different clinical groups in the embedding space. Since the learned DETree is in a high dimensional embedding space ($k = 5, 10, 15, 20, 25$), we adopted Principal Component Analysis (PCA) to project highdimensional features into two-dimensional space. PCA is a dimensionality reduction technique widely used in data analysis and machine learning. Its primary objective is to simplify complex datasets by transforming them into a lower-dimensional form while preserving the most critical information. The results are shown in Fig. 4.2 (a), where five subfigures (a1 to a5) correspond to the results of five runs of our experiments. From the visualization, it is evident that the feature distribution corresponding to the five different clinical stages exhibit a clear order (highlighted by dashed arrows), ranging from NC (green) to SMC (yellow), EMCI (orange), LMCI (pink), and eventually ends with AD (red), which is consistent with the AD progression trajectory. This demonstrates that the embedding space aligns well with the AD

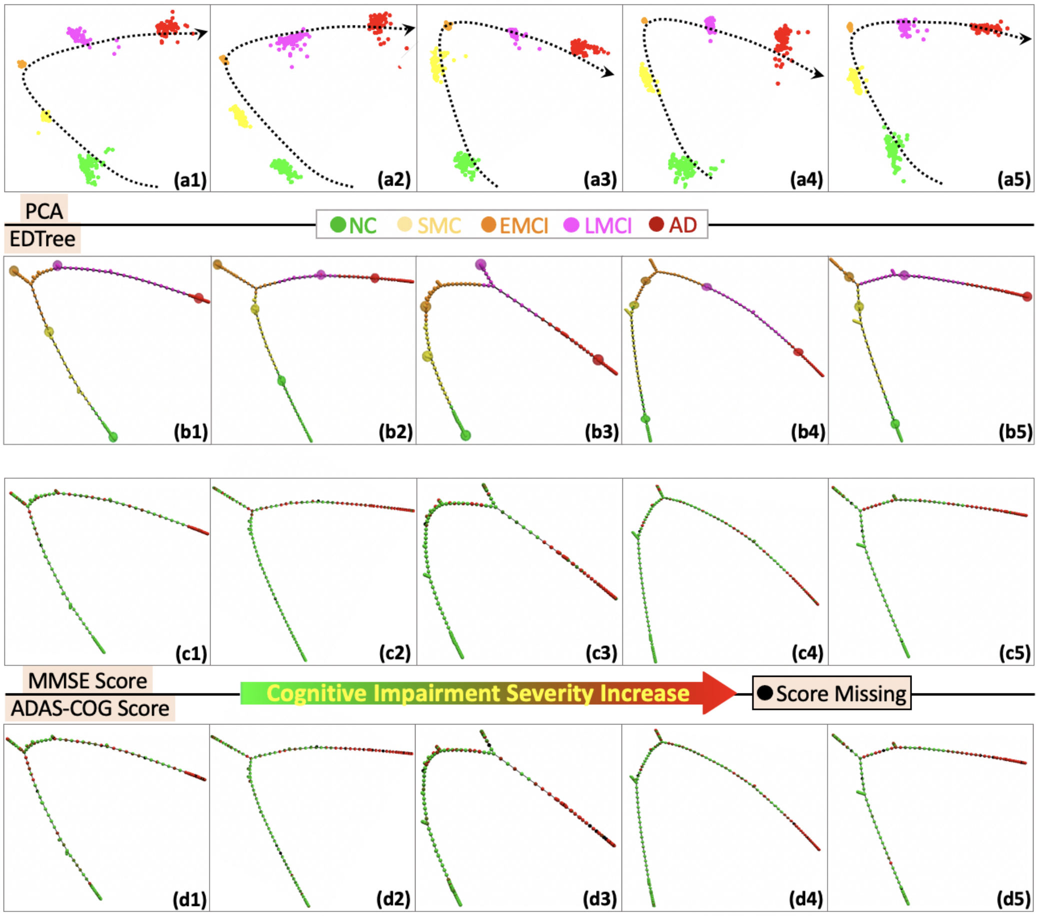


Figure 4.2. (a): Visualization of the feature distribution of different clinical groups in the embedding space. (b): DETree learned from multiple clinical groups, including NC, SMC, EMCI, LMCI, and AD. Each small bubble in the tree represents a single subject color-coded according to their clinical group. Each edge indicates higher similarity between connected nodes. The five larger bubbles represent the group level embeddings. (c): the Mini-Mental State Examination (MMSE) score [3] mapped to DETree. (d): the Alzheimer's Disease Assessment Scale – Cognitive subscale (ADAS-cog) score [4] mapped to DETree. The small bubbles in (c) and (d) correspond to the small bubbles in (b) at the same location.

progression. This result indicates that the proposed ordered embedding method is effective in capturing the progression of AD in the embedding space.

Secondly, we conducted analyses at the individual level. For each subject, the well-trained model mapped the input individual feature to the embedding space, obtaining the corresponding embedding vector $-h(x, \theta)$. To analyze the relationship between subjects, we employed Kruskal's algorithm to create a minimum spanning tree over the embedding vectors (DETree). The results are shown in Fig. 4.2 (b). In this tree structure, each small bubble represents a single subject, with its color indicating the clinical group to which the subject belongs. Each edge in the tree structure indicates higher similarity between the connected nodes. The five larger bubbles represent the learned group embeddings. From the results we can see that, the learned DETree structure precisely displays a trajectory of AD progression. It starts with the NC, goes through SMC, EMCI, LMCI and eventually ends with AD. The DETree captures the underlying progression pattern, demonstrating its ability to represent the continuous spectrum of AD progression in the embedding space. Moreover, a significant advantage of DETree over traditional classification methods is its ability to represent not only clinical group information but also the specific states of individuals along the entire disease development process. To further illustrate this, we mapped the Mini-Mental State Examination (MMSE) score [40] and the Alzheimer's Disease Assessment Scale - Cognitive subscale (ADAS-cog) score [41], two commonly used tools for assessing dementia, to DETree and presented the results in Fig. 4.2 (c) and (d), respectively. It is evident from the two score trees that the cognitive impairment severity shows an increasing trend from normal control (NC) to Alzheimer's disease (AD) in both score trees, in alignment with the disease progression. Furthermore, considerable variabilities exist in the impairment severity among individuals within the same clinical group. DETree's such capability to preserve substantial individuality in AD progression sets DETree apart as a significant advantage in disease modeling and enhances its potential utility in clinical applications and personalized medicine.

In this work, we used functional connectivity to learn the DETree. To further explore which functional connections contribute most to the learned tree structure,

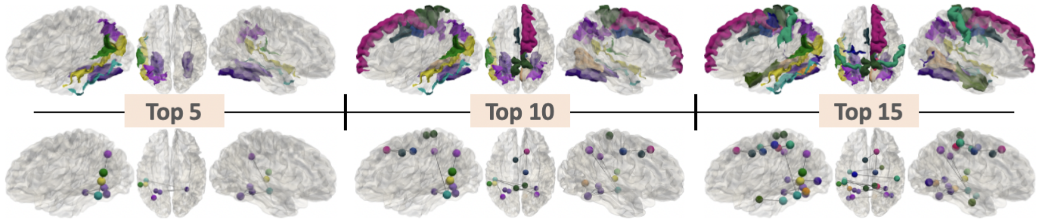


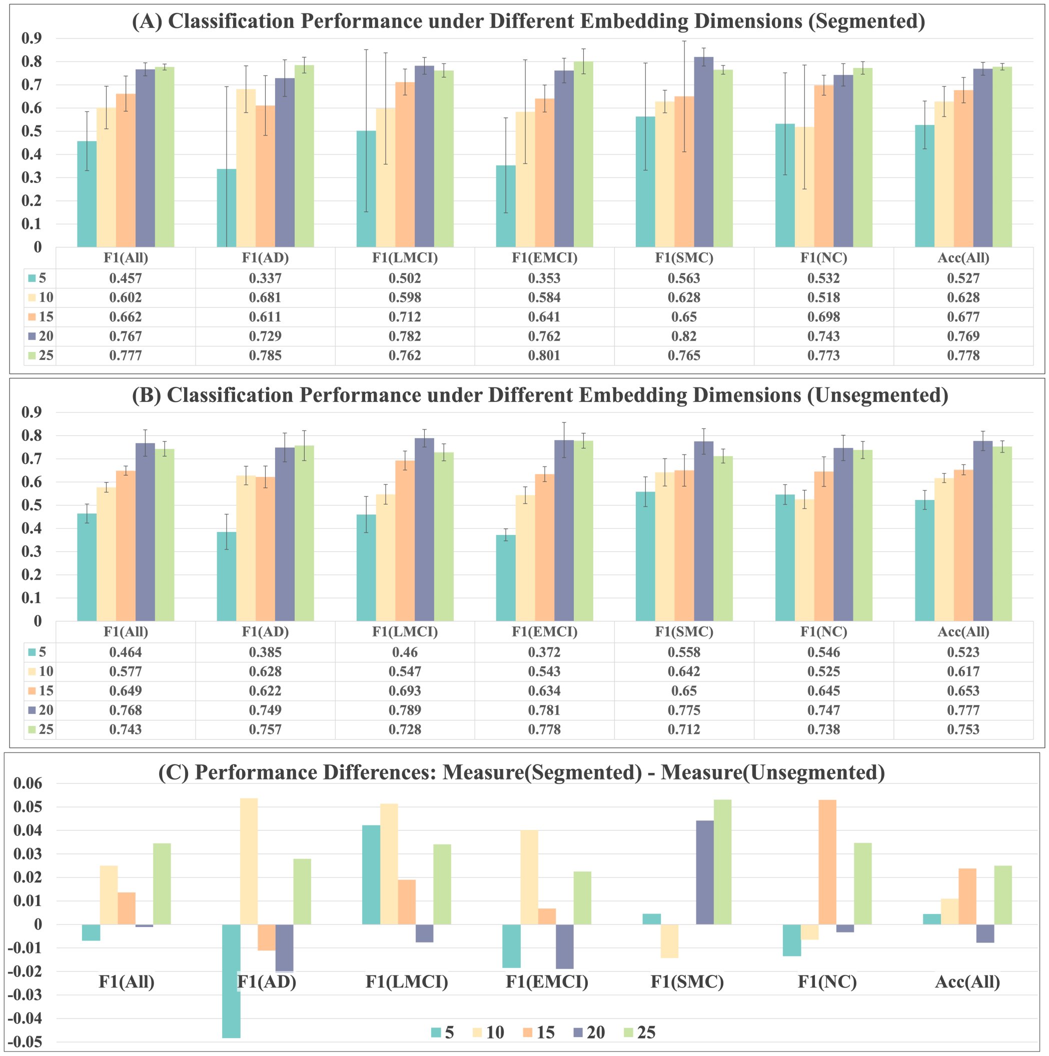
Figure 4.3. Top connectivity that contributes most to the learned DETree structure. In each block, the top and bottom rows display the involved brain regions and connectivity, respectively.

we sorted them with Laplacian score (LS) [146]. LS is a robust feature selection technique widely employed in machine learning and data analysis. It proves particularly useful when dealing with high-dimensional datasets and tasks that require a deep understanding of underlying data structures. LS is grounded in Laplacian Eigenmaps and Locality Preserving Projection, focusing on the evaluation of features based on their locality preserving power. The concept behind LS is straightforward: data points that belong to the same class are typically close to one another. LS uses the nearest neighbor graph to obtain the local structure of the data and obtains the LS value of each feature. Features that respect this graph structure will obtain higher LS values and will be selected. Fig. 4.3 shows the top 5, 10 and 15 connectivity selected by LS that have the most contributions during the learning of DETree structure. In each subfigure, the first row shows the brain regions involved in the connectivity. The second row shows the connectivity, and the corresponding regions are represented by bubbles with the same color. Most of the regions in Fig. 4.3 are reported in previous studies for the close relationship to AD, such as the regions in frontal lobe and temporal lobe [123, 41, 105].

4.3.4 Ablation Study

In our DETree model, the hyper-parameter that has the most significant influence on the DETree structure is k, which represents the dimension of the embedding space. We conducted experiments with different values of k, specifically $k = 5, 10, 15, 20$, and 25 , and analyzed the results from three perspectives: the classification performance, feature distribution in the latent space, and the learned DETree structure.

Firstly, we evaluated the influence of k on the classification performance. To augment the training dataset and improve model training, the fMRI signals of each individual in the training and validation datasets were divided into four non-overlapping segments, effectively quadrupling the dataset size. In order to assess whether the segmented methods impact the performance estimation on the testing dataset, we conducted experiments in both segmented and unsegmented settings on the testing datasets and compared the results. These comparisons are presented in Fig. 4.4. As depicted in Fig. 4.4(A) and (B), increasing the value of k leads to enhanced classification performance in both segmented and unsegmented settings on the testing dataset. This improvement is attributed to the fact that lowdimensional embedding spaces may not fully capture the intricate relationships within the brain network data, while higher-dimensional spaces provide a more comprehensive representation of these relationships, resulting in improved classification performance. To highlight the distinctions between the segmented and unsegmented testing dataset settings, we calculated the differences in F_1 and Acc measures and presented the results in Fig. 4.4(C). The findings indicate that segmented and unsegmented settings exhibit similar classification performance, with variations within the range of $[-0.05, 0.06]$.


(A) Classification Performance under Different Embedding Dimensions (Segmented)

	F1(All)	F1(AD)	F1(LMCI)	F1(EMCI)	F1(SMC)	F1(NC)	Acc(All)
5	0.457	0.337	0.502	0.353	0.563	0.532	0.527
10	0.602	0.681	0.598	0.584	0.628	0.518	0.628
15	0.662	0.611	0.712	0.641	0.65	0.698	0.677
20	0.767	0.729	0.782	0.762	0.82	0.743	0.769
25	0.777	0.785	0.762	0.801	0.765	0.773	0.778

(B) Classification Performance under Different Embedding Dimensions (Unsegmented)

	F1(All)	F1(AD)	F1(LMCI)	F1(EMCI)	F1(SMC)	F1(NC)	Acc(All)
5	0.464	0.385	0.46	0.372	0.558	0.546	0.523
10	0.577	0.628	0.547	0.543	0.642	0.525	0.617
15	0.649	0.622	0.693	0.634	0.65	0.645	0.653
20	0.768	0.749	0.789	0.781	0.775	0.747	0.777
25	0.743	0.757	0.728	0.778	0.712	0.738	0.753

(C) Performance Differences: Measure(Segmented) - Measure(Unsegmented)

Figure 4.4. (A) and (B): Classification performance of various dimensions of the embedding space in two testing dataset settings. (C): The differences in F_1 and Acc measures of two settings.

Secondly, we assessed the impact of k on the feature distribution in the embedding space, particularly its alignment with the AD progression process. The results are displayed in Fig. 4.5. Notably, as we varied the value of k from 5 to 25 , the feature distribution of the five clinical groups consistently maintained the order from NC to SMC, EMCI, LMCI, and eventually AD. This alignment of embedding space with the AD progression trajectory demonstrates that the choice of k does not affect the feature distribution in the embedding space. This indicates that our proposed ordered embedding method exhibits excellent robustness across different dimensions of embedding space. The flexibility to choose different dimensions of the embedding space enhances the potential of the proposed model in adapting to diverse data characteristics and complexities, making it applicable to a wide range of applications.

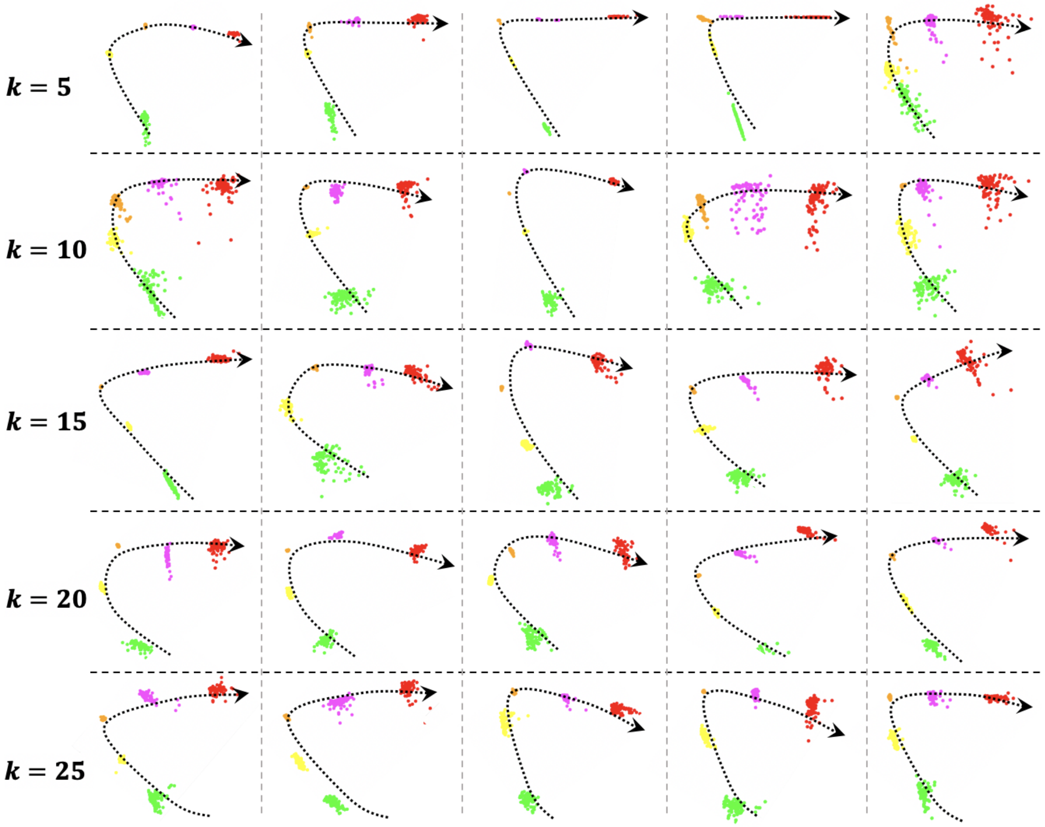


Figure 4.5. Visualization of the feature distribution of different clinical groups in the embedding space with varying dimensions: NC - green, SMC - yellow, EMCI - orange, LMCI - pink, and AD - red.

Thirdly, we explored the impact of k on the learned DETree structure and presented the results in Fig. 4.6. From Fig. 4.6 we can see that if k is too small (corresponding to lower dimensional embedding space), the distances among different embeddings tend to be small (highlighted by blue circle). As a result, the corresponding embeddings exhibit high similarities. This leads to insufficient dissimilarity between embeddings, which can limit the capability of DETree in representing multiple clinical stages in AD progression and compromise its prediction performance when estimating new samples. These findings are consistent with the results in Fig. 4.4, where lower-dimensional spaces exhibited inferior classification performance compared to higher-dimensional embedding spaces.

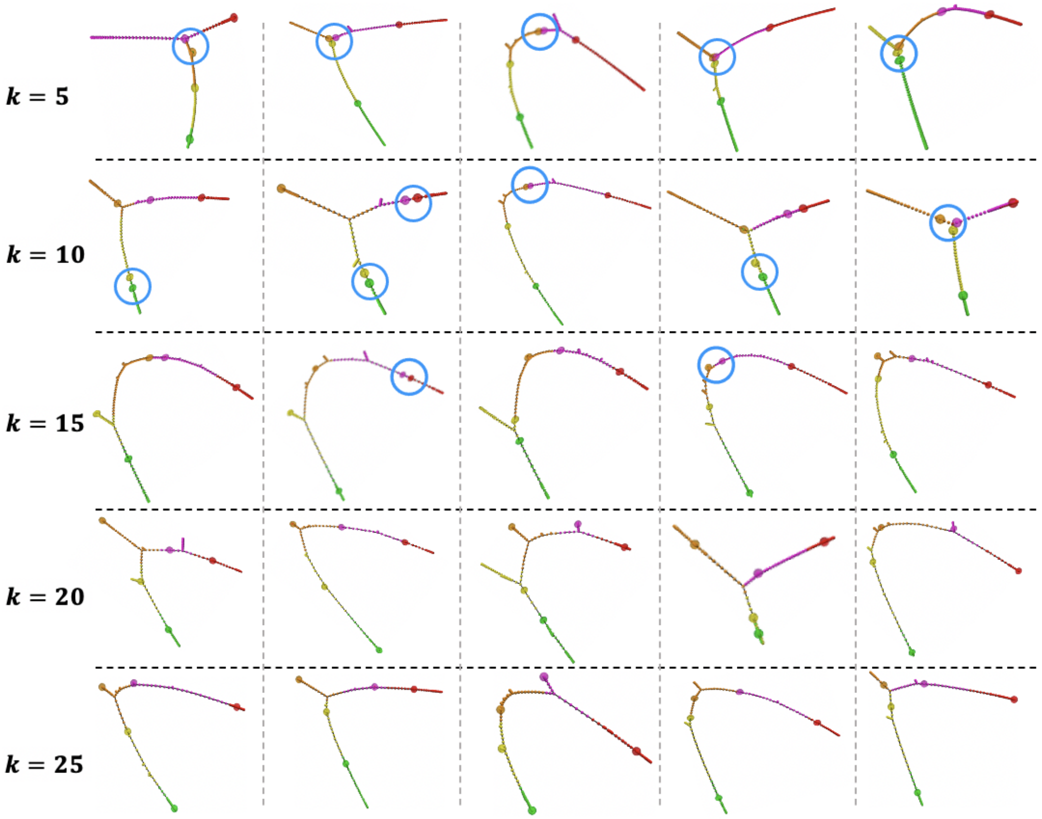
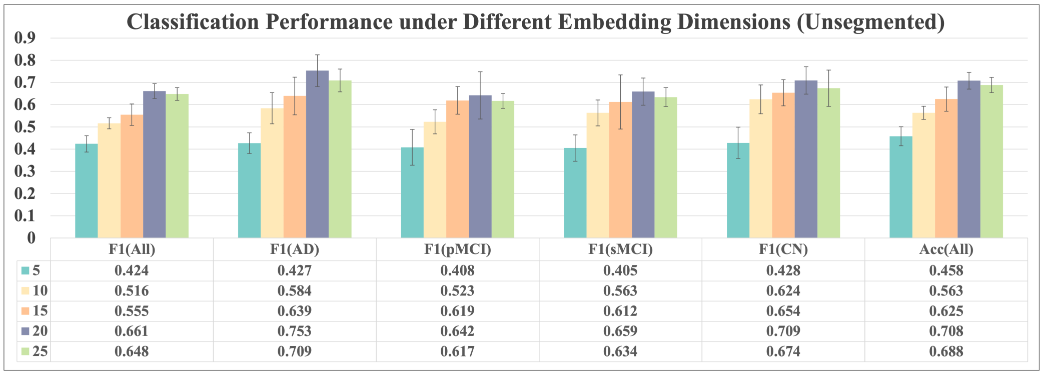


Figure 4.6. Different DETree structures learned in the embedding space with varying dimensions. The blue circles are used to highlight the embeddings from different clinical groups with small distance.

4.3.5 Reproducibility and Generalizability

To evaluate the reproducibility of our proposed model across different datasets and its generalizability to different tasks, we collected the most recent released subset of ADNI, including 145 subjects (40NC : 22 females, 18 males, 73.64 ± 7.03 yrs.; 28 progressive MCI (pMCI): 13 females, 15 males, 72.44 ± 7.43 yrs.; 42 stable MCI (sMCI): 18 females, 14 males, 71.32 ± 6.94 yrs.; and 35 AD: 15 females, 20 males, 72.93 ± 8.56 yrs.). The pMCI group includes patients who progressed to AD within 36 months [134, 135], while sMCI consists of individuals who did not progress. We divided the 145 subjects (dataset-2) into training, validation, and testing datasets. For the training and validation datasets, we employed the same four-segment approach as in dataset-1. However, in the testing dataset, we used an unsegmented setting. Using a similar experimental setup as in dataset-1, we conducted a series of experiments, repeating each experiment 5 times with different random seeds. Our objective was to evaluate the performance of the proposed DETree model with varying embedding dimensions (k) for the new classification task: NC vs. SMCI vs. pMCI vs. AD. We evaluated the model's performance from three perspectives: classification performance (Fig. 4.7), feature distributions in the latent space (Fig. 4.8), and the learned DETree structures (Fig. 4.9).

As shown in the results, increasing the value of k leads to improved classification performance, consistent feature distributions in the latent space, and larger distances among different embeddings within the learned DETree structure. These findings are consistent with the conclusions drawn from dataset-1, indicating that the DETree model exhibits excellent reproducibility and generalizability. However, it's worth noting that in the new classification task, our model achieved a maximum accuracy of 0.708 , slightly lower than the 0.777 accuracy obtained in task- 1 based on dataset-1 . This drop in accuracy can be attributed to the introduction of new categories,



	F1(All)	F1(AD)	F1(pMCI)	F1(sMCI)	F1(CN)	Acc(All)
5	0.424	0.427	0.408	0.405	0.428	0.458
10	0.516	0.584	0.523	0.563	0.624	0.563
15	0.555	0.639	0.619	0.612	0.654	0.625
20	0.661	0.753	0.642	0.659	0.709	0.708
25	0.648	0.709	0.617	0.634	0.674	0.688

Figure 4.7. Classification performance of various dimensions of the embedding space based on dataset2.

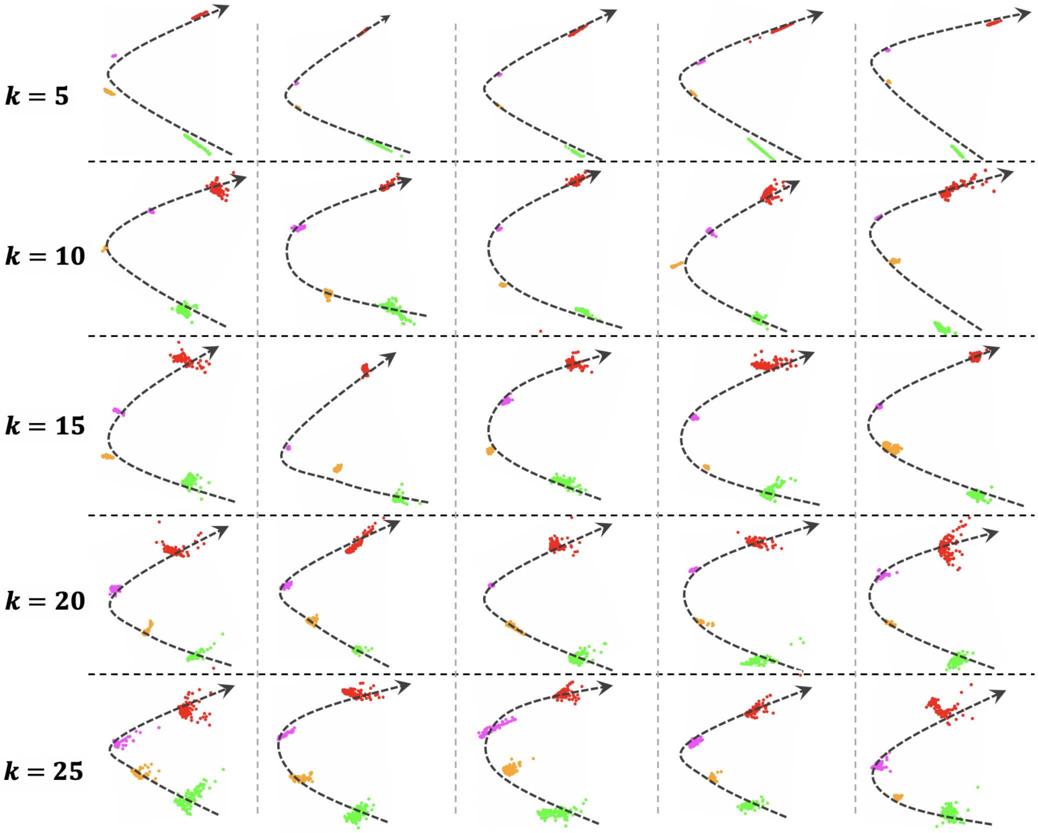


Figure 4.8. Visualization of the feature distribution of different clinical groups in the embedding space with varying dimensions based on dataset2.

sMCI and pMCI, which replaced EMCI and LMCI from task-1. This observation is consistent with the results in Table. 4.2, where tasks related to MCI progression, as

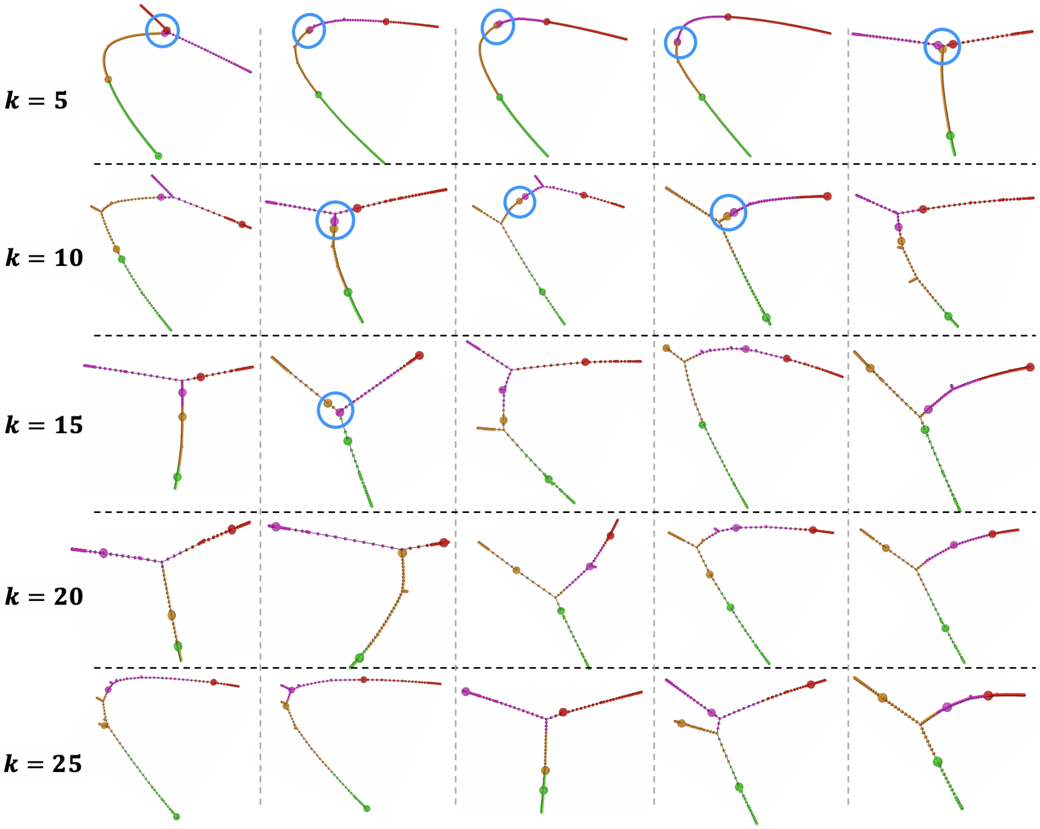


Figure 4.9. Different DETree structures learned in the embedding space with varying dimensions based on dataset2.

discussed in references [134, 135, 140], typically exhibit slightly lower classification accuracy compared to other works. This discrepancy may be due to the increased challenge of distinguishing between sMCI and pMCI compared to the previous EMCI and LMCI categories. Nevertheless, even in this context, our model consistently outperforms other models in the MCI progression task [134, 135, 140].

4.4 Discussion and Conclusion

In this study, we introduced a novel DETree framework that seamlessly integrates individual prediction with AD progression modeling. The learned DETree structure effectively represents the trajectory of AD progression and achieves an impressive prediction performance of over 77.8% for multiple AD-related stages. One of the key strengths of our approach is its ability to not only predict the clinical status of individual patients but also provide valuable information about their specific state within the entire spectrum of AD progression. We summarize the advantages and limitations of our current work and provide some insights for future research.

4.4.1 Advantages

DETree is a general framework for modeling continuous diseases development. In this work, we only applied DETree to Alzheimer's disease, but it is a versatile framework that can be extended to a wide range of diseases. The proposed DETree framework allows for flexibility in implementing nonlinear function $h(x, \theta)$, making it adaptable to any disease that exhibits multiple clinical stages during its development. Researchers can choose a suitable model architecture for $h(x, \theta)$ implementation and input relevant features into the model based on the specific disease and tasks at hand. More important, by modifying the affinity matrix $\mathcal{A}$, the prior knowledge about the disease can be easily introduced into the DETree model.

DETree exhibits versatility beyond the realm of classification tasks. With minor adjustments to the additional neural network layer in Section 4.2.4, DETree can seamlessly extend its applicability to regression problems. For instance, by substituting the discrete clinical labels with continuous clinical scores, such as the MMSE score [3], the existing classification framework can be readily transformed into a robust regression-based model. A straightforward approach to achieve this is as follows: We take the feature and score pairs $\{(x_i, s_i)\}_{i=1}^{n}$ as input to train a non-linear function $h(x, \theta) : \mathcal{R}^d \to \mathcal{R}^k$, with learnable model parameter θ. This transforms any given input $x \in \mathcal{R}^d$ to a latent space $\mathcal{R}^k$. Based on the latent feature $h(x, \theta)$, we will integrate two tasks: clinical score prediction and disease progression representation.

This means $h(x, \theta)$ is influenced by both tasks, allowing it to capture key information related to both scores and disease progression simultaneously. To implement the prediction task, we can train a submodule $g\left(x, W_r, b_r\right)$ to predict the clinical score as follows: ${s_i}' = g\left(h\left(x_i, \theta\right), W_r, b_r\right)$, and a regression loss $\mathcal{L}_r = \frac{1}{n}\sum_{i=1}^{n}\left(s_i - {s_i}'\right)^2$ can be used to control the regression learning. Regarding the disease progression task, we can utilize a ranking loss function [147]. In contrast to other loss functions like Cross-Entropy Loss or Mean Square Error Loss, which aim to directly predict a label, a value, or a set of values based on an input, Ranking Losses focus on predicting relative distances between inputs, commonly referred to as metric learning. Specifically, let $\mathbf{x} = \{x_1, \ldots, x_n\}$ be the objects that need to be ranked, associated with multi-level ratings denoted as $L = \{l(1), \ldots, l(n)\}$, where each $l(i) \in \{r_1, \ldots, r_k\}$ and signifies the label of x_i [147]. In our work, the rating $l(i)$ is the clinical score of x_i. In the ranking context, when $l(i) > l(j)$, it implies that object x_i should take precedence over x_j in the ranking order. Let $\mathcal{F}$ represent the class of functions, and $f \in \mathcal{F}$ denote a specific ranking function. The objective is to learn the optimal ranking function from training data by minimizing a specific loss function. This loss function is defined based on the objects, their associated labels, and the ranking function itself. Several methodologies have been proposed to facilitate the learning of this optimal ranking function, including point-wise approaches [148, 149], pairwise approaches [150, 151], and e list-wise approaches [152, 153]. Moreover, it's worth delving deeper into exploring the integration of disease-related prior knowledge to tailor a ranking function that better aligns with the demands of our specific task. This is an avenue that deserves further investigation in our future work.

4.4.2 Limitations and Future Work

The current work has focused on a specific choice of the embedding number $-K$, for each class. In Section 4.2.3, we provide a general form of group embeddings as $\varepsilon = \left\{ e_{i,j} \in \mathcal{R}^k \mid i = 1, 2, \cdots, C; j = 1, 2, \cdots, K \right\}$ where K is the number of the embeddings in each class and could be equal to or greater than 1 . However, for the particular application addressed in this study, K is intentionally set to 1 . It's important to note that in many scenarios, the choice of K can be a subject for more in-depth exploration. For example, in the case of specific diseases where individuals in the same disease stage exhibit diverse clinical symptoms, the choice of K becomes crucial. In such cases, the objective is that each of the K learned embeddings will capture the specific features associated with the varying clinical symptoms observed at the same disease stage. It's essential to emphasize that, given the potential subtle differences among patients in the same disease stage, a significant amount of data is required to adequately train the model for each clinical response.

Current work only covers single modality. Due to the scarcity of multi-modal data, where each subject is required to have multiple modalities of data, such as both structural and functional image data, the sample size drastically decreases. Consequently, we have only focused on utilizing a single modality for this study. Previous studies [154, 155] suggest that functional abnormalities may precede structural alterations, making functional data more suitable for a comprehensive modeling of the entire AD progression process. Therefore, we chose to use functional connectivity data in this study. In the future, as we acquire sufficient multi-modal data, extending our current model to accommodate multiple modalities becomes feasible by treating each modality as a distinct view. Specifically, we can tailor a modality-specific model for each modality, mapping the original data into a unified embedding space from its unique perspective. This flexible strategy not only effectively conceals heterogeneities and inconsistencies between modalities, such as variations in numerical values, dimensions, and representations, but also allows us to optimize the model architecture for each modality to capture its specific characteristics most effectively. For instance, recent advancements in Large Language Models (LLMs) have demonstrated impressive achievements in various domains [156, 157, 158, 159, 160], including NLP and Computer Vision (CV). We can leverage these pre-trained LLMs on large datasets as feature extractors and fine-tune them to adapt to healthcare data. This enables us to fully exploit the remarkable generalization and feature extraction capabilities of these large models. Through this approach, we can harness cutting-edge models and technologies from the NLP and CV domains to study brain diseases effectively. By integrating NLP and CV advancements into our research on brain diseases, we have the opportunity to gain new insights and breakthroughs that were previously not attainable. This interdisciplinary approach opens up exciting possibilities for the advancement of neuroscience and healthcare research.

CHAPTER 5

CORTEX2VEC: ANATOMICAL EMBEDDING OF CORTICAL FOLDING

In **Chapter 5**, we will explore how to leverage and customize the important techniques – tokenization and embedding in LLMs, to address the longstanding anatomical correspondence problem in neuroscience. Currently, brain correspondence approaches highly depend on the regularity, or commonality, of anatomical structure, by forcing the same atlas to be matched to different brains. As a result, individualized structural information can be overlooked. Recently, we conceptualized a new type of cortical folding pattern called the 3-hinge gyrus (3HG), which is defined as the conjunction of gyri coming from three directions. Many studies have confirmed that 3HGs are not only widely existing on different brains, but also possess both common and individual patterns. In our study, we put further effort, based on the identified 3HGs, to establish the correspondences of individual 3HGs. Inspired by the embedding technique in LLMs as well as taking into account the unique characteristics of brain networks, we developed a learning-based embedding framework to encode individual cortical folding patterns into a group of anatomically meaningful embedding vectors (cortex2vector). Each 3HG can be represented as a combination of these embedding vectors via a set of individual specific combining coefficients. In this way, the regularity of folding pattern is encoded into the embedding vectors, while the individual variations are preserved by the multi-hop combination coefficients. Results show that the learned embeddings can simultaneously encode the commonality and individuality of cortical folding patterns, as well as robustly infer the complicated many-to-many anatomical correspondences among different brains.

5.1 Introduction

Accumulating evidence suggest that the underlying mechanisms of brain organization are embedded in cortical folding patterns [161, 162, 163, 164, 165]. Alterations or deficits in cortical folding are strongly associated with abnormal brain structure-function, impaired cognition, behavioral disorganization [166, 165, 94] and various neurodevelopmental disorders [167, 168, 169]. Unfortunately, quantitative, and effective representation of cortical folding has been challenging due to the remarkable complexity and variability of convex gyral/concave sulcal shapes. Recently, we first conceptualized a new type of brain folding pattern termed 3-hinge gyrus (3HG) [170, 171], which is the conjunction of gyri coming from three directions in cortical folding. Interestingly, 3HGs are not only evolutionarily preserved across multiple species of primates [172], but also robustly existed on human brains despite different populations or brain conditions [171, 173, 174, 175]. Previous studies already confirmed that 3HGs tend to have thicker cortices [170], higher DTI-derived fiber density [173], and more pronounced connective diversities in both structural and functional domains [170, 172, 174]. In addition, comparing to other gyral regions (i.e., 2-hinge gyrus), 3HGs possess significantly higher brain connectivity measures [174] such as degree, strength and betweenness. All these findings suggest that 3HGs may play a key role (e.g., hubs in the cortico-cortical connective network) in brain anatomical architecture [174]. Meanwhile, 3HGs can be achieved conveniently from widely existing T1 images and therefore, can better serve as meso-scale anatomical landmarks of human brain.

It is worth noting that 3HGs are identified on individual space, which means cross-subject correspondences of 3HGs need to be constructed before conducting any population-level analysis. Zhang et al. [175] developed a two-view & group-wise graph matching method to use both cortical folding patterns and DTI-derived fiber shape features to estimate the 3HG correspondences. The core idea is to jointly optimize the axonal connective and anatomical topological patterns, as two views, and maximize the consistency between the corresponding 3HGs on different brains. However, this method suffered from three challenges: 1) because of group-wise optimization scheme, the robustness and computing time highly rely on the number of samples; 2) the graph matching is conducted independently (from scratch), making it difficult to generalize the obtained correspondence of 3HGs on training data to new brains; and 3) the features of two views are handcrafted and directly used to seek the consistency, thus weakening the effectiveness of the method for inflexibility to adapt tremendous individual variations. Therefore, it is more desirable to automatically learn an intrinsic representation of folding patterns that can be used for finding reliable corresponding 3HGs across different brains. More importantly, this representation should be able to simultaneously characterize commonality and individuality of cortical folding and be generalized well on new datasets.

Recent advances in deep modeling have triggered a new era in representation learning field, and a variety of powerful representation learning algorithms have been proposed. For example, in Natural Language Processing (NLP), many word embedding methods have been developed to learn semantically meaningful embeddings. Those embeddings have shown superior performances on various downstream tasks [176, 177, 178, 179]. Similarly, there has been a surge of graph-based embedding approaches that can encode the nodes/edges based on graph structure information [180, 181, 182, 183, 184, 185]. All these remarkable works have demonstrated the superiority of learning-based embedding methods when targeting an effective representation in latent space. Inspired by these representation learning studies, in this work, we aim to design a learning-based embedding framework to encode the complex and variable cortical folding patterns into a group of anatomically meaningful embedding vectors (cortex2vector). The similarity between different embedding vectors, in turn, can appropriately represent the relations between the corresponding brain landmarks – 3HGs.

In word embedding, each single word in a sentence can be explicitly embedded via alphabet combinations and this makes it easy to build the word vocabulary. When using 3HGs as the elementary anatomical units for cortical embedding, we do not have similar "vocabulary", since every single 3HG is unique. To solve this problem, instead of conducting embedding on 3HG directly, we choose to learn the embeddings of the anatomical features associated to 3HGs. Our previous work [171] has developed an innovative and effective algorithm to build brain anatomical graph, named GyralNet, that automatically and accurately extracts all gyral crest lines as edges, and 3HGs as nodes. Hence, for each 3HG, its location and the connections with other 3HGs within individual GyralNet can be used as two key features for embedding. Specifically, we used the anatomical regions of interest (ROI) (from FreeSurfer atlas) to index the location of each 3HG. As all the 3HGs on the same hemisphere are connected by gyri hinges, we considered multi-hop neighbors of each 3HG and build the local connections. Though this way, each 3HG can be represented as a hierarchical combination of multi-hop ROIs via a set of specific multi-hop combination coefficients. That is, we learned a high-dimensional embedding vector for each anatomical ROI via an autoencoder model, and these learned ROI embedding could serve as the basic elements to represent each 3HG, like alphabet in words. By training the proposed model in a self-supervised manner, the regularity of folding pattern is encoded into the embedding vectors and the variability is preserved by the multi-hop coefficients. Our experiment results show that the learned embeddings can successfully encode the common patterns and variability of 3HGs simultaneously and can also accurately infer the cross-subject many-to-many correspondence under complex cortical landscapes.

Moreover, our developed learning-based framework generalizes well when applied to large-scale new datasets.

5.2 Materials and Methods

5.2.1 Datasets Description and Data Pre-processing

In this work, we used structure MRI (T1-weighted) of 1064 subjects from Human Connectome Project (HCP) S1200 release. For the T1-weighted structure MRI, the imaging parameters are $\mathrm{TR} = 2.4s, \mathrm{TE} = 2.14$ ms, flip angle = 8deg, Field of View $(FOV) = 224 \times 224$ mm and resolution $= 0.7 \times 0.7 \times 0.7$ mm^3. We applied the same standard pre-processing procedures as in references [144] for T1 imaging data. In brief, pre-processing steps included brain skull removal, tissue segmentation and cortical surface reconstruction via FreeSurfer package [49]. Destrieux Atlas [1] was used to conduct ROI labeling for reconstructed white matter surface.

5.2.2 Identification of GyralNet and 3HGs

The 3HGs were identified automatically via our recently developed pipeline [171], which consists of four key steps, including gyral altitude mapping, gyral crest segmentation, tree marching and extraction of GyralNet and 3HGs.

5.2.2.1 Gyral Altitude Mapping

The gyral altitude is defined as the displacement of a vertex from its original location to a hypothetical "mid-surface", which separates gyri from sulci [186]. This "mid-surface" is chosen to make the summation of the displacement of all vertices from their original locations to be zero. We mapped gyral altitude to all vertices on surface in Fig. 5.1(a1).

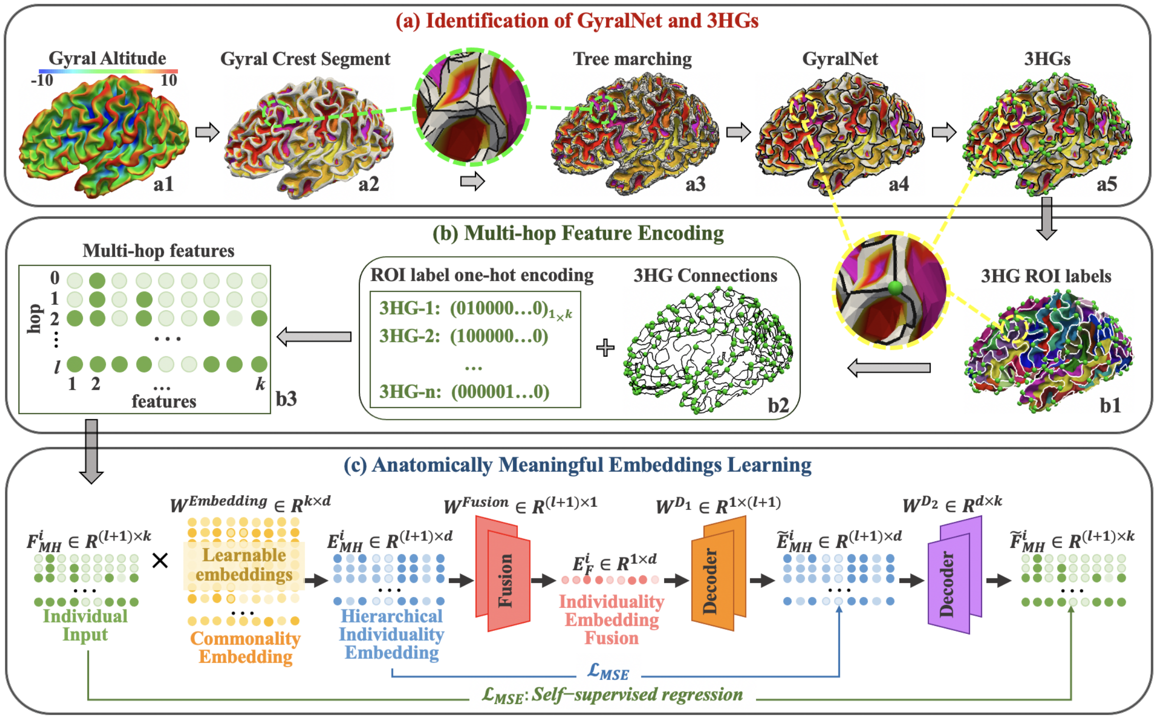

(a) Identification of GyralNet and 3HGs
Gyral Altitude
-10
10
a1
Gyral Crest Segment
a2
Tree marching
a3
GyralNet
a4
3HGs
a5
(b) Multi-hop Feature Encoding
Multi-hop features
hop
0
1
2
l
1
2
features
k
b3
ROI label one-hot encoding
3HG-1: $(010000\ldots0)_{1\times k}$
3HG-2: $(100000\ldots0)$
3HG-n: $(000001\ldots0)$
3HG Connections
b2
3HG ROI labels
b1
(c) Anatomically Meaningful Embeddings Learning
$F^i_{MH} \in R^{(l+1)\times k}$
Individual Input
$W^{Embedding} \in R^{k\times d}$
Learnable embeddings
Commonality Embedding
$E^i_{MH} \in R^{(l+1)\times d}$
Hierarchical Individuality Embedding
$W^{Fusion} \in R^{(l+1)\times 1}$
Fusion
$E^i_F \in R^{1\times d}$
Individuality Embedding Fusion
$W^{D_1} \in R^{1\times(l+1)}$
Decoder
$\widetilde{E}^i_{MH} \in R^{(l+1)\times d}$
$W^{D_2} \in R^{d\times k}$
Decoder
$\widetilde{F}^i_{MH} \in R^{(l+1)\times k}$
$\mathcal{L}_{MSE}$
$\mathcal{L}_{MSE}$: Self–supervised regression

Figure 5.1. Schematic of the proposed embedding framework. **(a)** Identification of GyralNet and 3HGs. **a1** White matter cortical surface mapped by gyral altitude; **a2** Segmentation of gyral crest (white regions) from sulci basins (color regions); **a3** Connecting the gyral crest regions into a completed graph by tree marching (black curves). A magnification view of the circled patch is displayed between a2 and a3; **a4** Pruning the redundant branches to preserve the main trunk of the graph (black curves) - GyralNet; **a5** Identification of 3HGs (labeled by green bubbles). **(b)** 3HG's multi-hop feature encoding. **b1** Parcellating the entire cortex into 75 ROIs via Destrieux Atlas and assigning each 3HG with an ROI label as node feature; **b2** Numerically representing each ROI label by one-hot encoding; **b3** By considering multi-hop neighbors, 3HG are encoded by multi-hop features. **(c)** The proposed learning-based embedding framework (details can be found in Section 5.2.4).

5.2.2.2 Gyral Crest Segmentation

The watershed algorithm (Bertrand (2005)) was applied to the gyral altitude map in Fig. 5.1(a1) to separate the gyral crest (regions above a predefined gyral altitude level) from the sulcal basins (regions below the altitude level). The obtained gyral crests (white) and sulcal basins (labeled by different colors) were displayed in

Fig. 5.1(a2). More details about the watershed segmentation progress can be referred to [171].

5.2.2.3 Tree Marching

A distance transform was firstly performed to gyral crest regions to assign a distance value to each vertex and highlight the centers of the gyral crests. The distance was defined based on the displacement from the vertex of interest to the boundaries between gyral crest regions and sulcal basins. As a result, a field with decreasing gradient from gyral crest centers to the boundaries was generated. Then a tree marching algorithm was applied to the distance map to connect vertices from crest centers to boundaries. A tree root was placed in each gyral crest center and progressively connected other vertices following the descending gradient of the distance map till the boundaries between gyral crest regions and sulcal basins are reached. During this process, when two trees met, connections will be made between the two trees. By this way, the gyral crests on the same hemisphere can be connected into a graph. The constructed graph structure was shown in Fig. 5.1(a3) by black curves and the zoom in view of a circled area was displayed between Fig. 5.1(a2) and Fig. 5.1(a3).

5.2.2.4 Extraction of GyralNet and 3HGs

During the tree marching process, all the vertices in gyral crest regions were connected and some redundant branches were generated. We trimmed these redundant branches when their length was shorter than a predefined threshold. The main trunks of the graph structure were preserved, and this trimmed graph named as GyralNet (black curves in Fig. 5.1(a4)). The conjunctions with three branches on the GyralNet were defined as 3HGs (green bubbles in Fig. 5.1(a5)).

5.2.3 Multi-Hop Features Encoding

By taking 3HGs as nodes, the GyralNet can be represented as an undirected graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the undirected graph, where $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ is the set of N HGs, and $\mathcal{E} \subseteq \{\{v_i, v_j\} \mid v_i, v_j \in \mathcal{V}\}$ is the set of unweighted edges which are the gyral crest lines connecting 3HGs. Its adjacency matrix is denoted by $\boldsymbol{A} = [a_{i,j}] \in \boldsymbol{R}^{N\times N}$, where $a_{i,j} = 1$ if there is a connection between v_i and v_j, and $a_{i,j} = 0$ otherwise. We conducted ROI labeling via Destrieux Atlas and divide the whole surface into 75 ROIs. Each 3HG was assigned an ROI label as node feature (Fig. 5.1(b1)). We numerically represented the ROI labels by one-hot encoding, i.e., the kth label was denoted by $\boldsymbol{e_k} \in \boldsymbol{R}^{75}$ with 1 in the kth location and 0 elsewhere. Accordingly, the ith 3HG in k^{th} ROI can be denoted by $\boldsymbol{x_i} = \boldsymbol{e_k}$.

By far, the undirected graph of 3HGs can be represented by the adjacency matrix $\mathcal{A}$ and the feature matrix $\boldsymbol{X} = \{\boldsymbol{x_1}; \boldsymbol{x_2}; \cdots; \boldsymbol{x_N}\} \in \boldsymbol{R}^{N\times 75}$ (Fig. 5.1(b2)). Based on the two matrices, the two key components of multi-hop features encoding are defined as:

l^{th} hop features: In a 3HG graph, the lth hop neighborhood of the 3HG is the set of 3HGs connecting to 3HG i via the shortest path with l steps (l-hop), denoted by $N_l(i)$. For $v_j \in N_l(i)$, its feature vector is denoted by $\boldsymbol{x_j}$, and accordingly the lth hop features of 3HG are defined as $\sum_{v_j \in N_l(i)} \boldsymbol{x_j}$. Given the adjacency matrix $\mathcal{A}$ and the feature matrix $\boldsymbol{X}$, the lth hop features of 3HG i can be calculated by $\left[\mathcal{A}^l \boldsymbol{X}\right]_{i,*}$, where $\mathcal{A}^l$ is the lth power of $\mathcal{A}$. As the adjacency matrix $\mathcal{A}$ defines the direct connections between the graph nodes, in the process of recurrently multiplying by itself, just like the graph convolution operation, the undirected connections of further neighbors are propagated and gathered along with the direct connections. When multiplying $\mathcal{A}$ l times, the features of the neighbors that can reach the center node by l steps are congregated. As each row of $\mathcal{A}^l\boldsymbol{X}$ corresponds to one 3HG, the lth hop feature of 3HG i thereby can be denoted by $\left[\mathcal{A}^l\boldsymbol{X}\right]_{i,*}$.

***l* hop features (multi-hop features):** Based on the definition of lth hop features, we further defined the l hop features of 3HG i as follows (Fig. 5.1(b3)):

$$F_{MH}^{i} = \begin{bmatrix} [\mathcal{X}]_{i,*} \\ [\mathcal{A}^1\mathcal{X}]_{i,*} \\ \vdots \\ \left[\mathcal{A}^{l-1}\mathcal{X}\right]_{i,*} \\ \left[\mathcal{A}^{l}\mathcal{X}\right]_{i,*} \end{bmatrix} \in R^{(l+1)\times 75}(5.1)$$

where the 0th hop feature $[\boldsymbol{X}]_{i,*} = e_k$, which indicates that 3HG i is in the ROI k, and the lth hop feature $\left[\mathcal{A}^l\boldsymbol{X}\right]_{i,*} = \sum_{k=1}^{75} a_{lk}^{i} e_k$, with the multi-hop coefficient a_{lk}^{i} indicating the number of different l step paths that are available from 3HG i to the ROI k. If there is no l-step path between them, $a_{lk}^{i} = 0$, otherwise, a_{lk}^{i} will be a positive integer. In this work, we set $l > 1$, hence the l hop features cover multiple hops of the 3HG graph and are also called multi-hop features. By organizing the multi-hop features in this manner, the hierarchical multi-hop relationships between 3HGs with all the 75 ROIs are encoded into F_{MH}.

5.2.4 Learning-based Embedding Framework

Our learning-based embedding framework (Fig. 5.1(c)) is designed in a self-supervised manner: it includes two-stage encoding to hierarchically map the input multi-hop features to a latent representation, and a two-stage decoding that aims to hierarchically reconstruct the original input from the latent representation. The embedding learning process can be formulated as (5.2):

$$\begin{aligned}
E_{MH}^{i} &= \sigma\left(F_{MH}^{i} \cdot W^{\text{Embedding}}\right) \\
E_{F}^{i} &= \sigma\left(\left(W^{\text{Fusion}}\right)^{T} \cdot E_{MH}^{i}\right) \\
\tilde{E}_{MH}^{i} &= \sigma\left(\left(W^{D_1}\right)^{T} \cdot E_{F}^{i}\right) \\
\tilde{F}_{MH}^{i} &= \tilde{E}_{MH}^{i} \cdot W^{D_2}
\end{aligned} \tag{5.2}$$

where σ is the non-linear activation function, $F_{MH}^{i} \in R^{(l+1)\times 75}$ is the multi-hop feature of 3HG i defined by (5.1); $W^{\text{Embedding}} = \{w_1; w_2; \cdots ; w_{75}\} \in R^{75\times d}$ is the learnable embedding matrix. In our setting, there are 75 ROIs in total, we initialized a learnable embedding vector $w \in R^{d}$ for each ROI and organized them as the same order as the one-hot encoding in (5.1) to form $W^{\text{Embedding}}$. The input multi-hop features are embedded via $W^{\text{Embedding}}$ (hop by hop) to generate multi-hop embeddings $E_{MH}^{i} \in R^{(l+1)\times d}$. To further fuse the multi-hop embeddings into a single embedding vector that contains the complete multi-hop information, we conducted the second encoding by learnable combination parameters $W^{Fusion} \in R^{(l+1)\times 1}$ to integrate the multi-hop embeddings into one embedding vector $E_{F}^{i} \in R^{1\times d}$. In (5.2) the transpose of $W^{\text{Fusion}} - \left(W^{\text{Fusion}}\right)^{T}$, was used for matrix multiplication at multi-hop dimension (rows of E_{MH}^{i}).

We used a symmetric design (as traditional autoencoder) for the two-stage decoding with the parameters $W^{D_1} \in R^{1\times (l+1)}$ and $W^{D_2} \in R^{d\times 75}$, respectively. The first decoding reconstructs the hierarchical multi-hop embeddings from the combined embedding vector, which ensures that the combined embedding vector E_{F}^{i} has captured the complete information to restore the embeddings for each hop. Then, upon the restored multi-hop embeddings $\widetilde{E}_{MP}^{i}$, the second decoding was applied to recover the multi-hop input features $\tilde{F}^i_{MH}$. We adopted the MSE loss to evaluate the two-stage decoding and the objective function can be defined by (5.3):

$$\mathcal{L} = \alpha \left\| E^i_{MH} - \tilde{E}^i_{MH} \right\|^2_F + \beta \left\| F^i_{MH} - \tilde{F}^i_{MH} \right\|^2_F \quad (5.3)$$

where α and β are the hyper-parameters to control the contribution of the two-stage decoding. The entire model was trained in a self-supervised manner, which avoids introducing any bias from supervised term when representing cortical folding as embedding vectors. Through a two-stage encoding-decoding process, the intrinsic patterns buried in the complex and variable cortical folding can be effectively instilled into the embedding vectors.

5.2.5 Evaluation of the Embedding Effectiveness

In this work, there are two kinds of embeddings: the ROI embeddings learned by population $W^{\text{Embedding}}$ and the individual 3HG embeddings $-E^i_F$. The ROI embeddings are learned by recovering the 3HG's multi-hop features on population level and can be served as the basic elements to represent each 3HG. Therefore, they should have the capability to characterize regularity of the anatomical pattern of 3HGs shared by the population. We adopted the strength of 3HG's multi-hop connection to describe the 3HGs' connection patterns between ROI pairs at different hop levels, which is defined as:

Strength of 3HG's multi-hop connection (ground truth matrix): If 3HG i is in ROIk and3HG j is in ROI m, and they are connected in the GyralNet via the shortest path with l steps, then ROI k and ROI m have one l-hop 3HG connection. In this work, we added up the connections between each pair of ROIs across the whole population and used the resulting matrix as the ground truth (Fig. 5.2 a).

In the ground truth matrix, if two ROIs have a larger number of l-hop 3HG connections, it means the two ROIs are closely related in GyralNet at the l-hop level. Thus, their l-hop embedding vectors should capture this close relationship by possessing higher similarity in the latent space.

The individual 3HG embedding E_F^i is used to represent a unique 3HG. As defined in (5.1) and (5.2), each 3HG embedding is a specific combination of ROI embeddings $W^{\text{Embedding}}$ via a set of 3HG specific coefficients $\{a_{lk}^i\}$ and the learned fusion parameters W^{Fusion}. An effective embedding vector E_F^i is expected to be able to preserve the individuality of different 3HGs and provides reliable crosssubject 3HG anatomical correspondence. To evaluate this capability, we applied the generated 3HG embeddings to the anatomical correspondence task to infer the complicated many-to-many crosssubject anatomical correspondence of 3HGs.

5.3 Results

We applied the proposed multi-hop feature encoding method (Section 5.2.3) and the learning-based embedding framework (Section 5.2.4) to the identified 3HGs (Section 5.2.2). By training the model end-toend in a self-supervised task, we learned a set of ROI embeddings $-W^{\text{Embedding}}$. The effectiveness of $W^{\text{Embedding}}$ was evaluated by the strength of 3HG's multi-hop connection (Section 5.2.5). Then we generalized the learned ROI embeddings and the well-trained model to a new dataset and generated an individual embedding vector for each 3HG. The effectiveness of the generated individual embedding vectors was evaluated in the anatomical correspondence task to infer the complicated cross-subject anatomical correspondence of 3HGs. The result section is organized as follows: Section 5.3.1 introduces the experimental setting; Section 5.3.2 evaluates the effectiveness of the learned ROI embeddings; Section 5.3.3

show the inference results of the 3HGs' anatomical correspondence task; Section 5.3.4 assesses the regression performance of the proposed two-stage decoding framework.

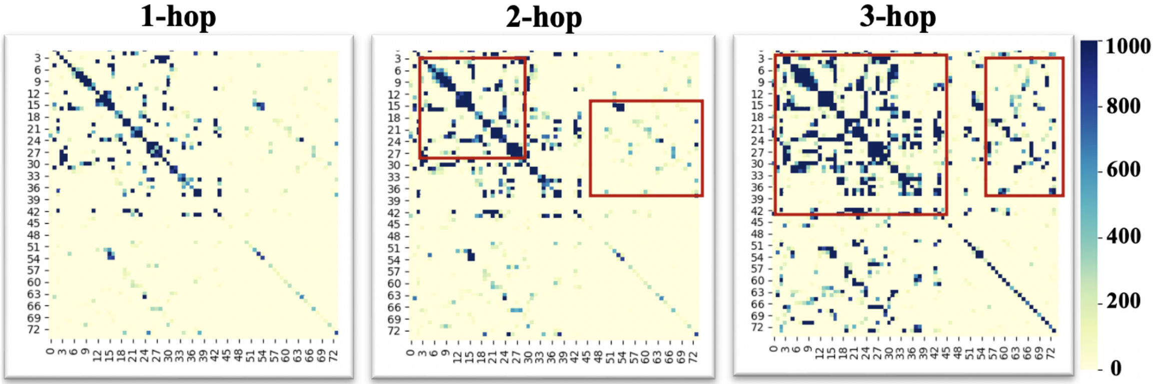
5.3.1 Experimental Setting

Data Setting. We randomly divided the 1064 subjects from HCP dataset into two datasets. Dataset-1 is used to train the model and learn the embeddings. Then the well-trained model and the learned ROI embeddings are applied to dataset-2 to infer the 3HGs' anatomical correspondence. In dataset-1, there are 564 subjects, and 186,915 3HGs are identified. In dataset-2, there are 500 subjects and 169,923 3HGs are identified. Each 3HG is treated as a data sample.

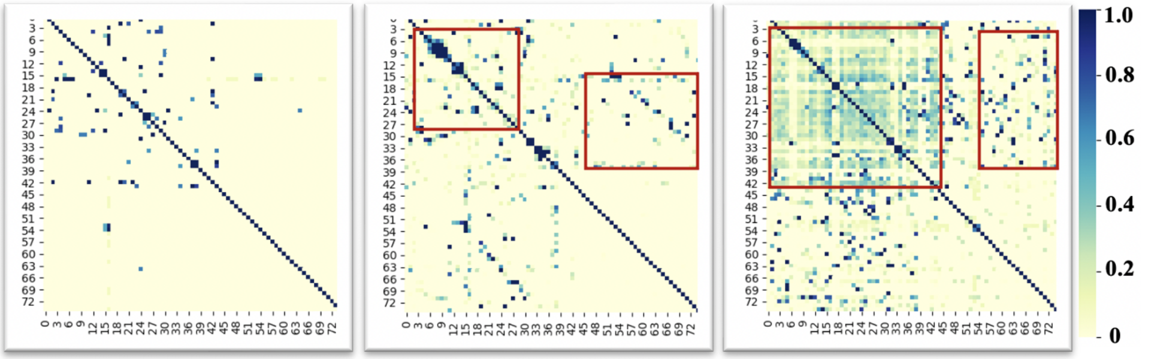
Model Setting. For multi-hop features, we generated 1-hop features ($l = 1$ in (5.1) and (5.2)), 2-hop features ($l = 2$) and 3-hop features ($l = 3$). For each kind of feature, we trained the model to learn the corresponding ROI embeddings. In our experiments, the learnable ROI embeddings were initialized by identity matrix to ensure the initial distances between any two embedding vectors are the same. We adopted the embedding dimension $d = 128$. The fusion operation $-W^{\text{Fusion}}$, and the two decoder operations $-W^{D_1}$ and W^{D_2} were implemented by fully connected layers and the parameters were initialized following the Xavier scheme. The entire model was trained in an end-to-end manner. The Adam optimizer was used to train the whole model with standard learning rate 0.001 , weight decay 0.01 , and momentum rates $(0.9, 0.999)$.

5.3.2 Effectiveness of ROI Embeddings

In the experiments, we used dataset-1 to generate different multi-hop features including 1-hop features, 2-hop features, and 3-hop features to train the model and learn the ROI embeddings. We evaluated the learned ROI embeddings via the



(a) ground truth matrices (strength of 3HGs' multi-hop connection)



(b) embedding similarity matrices

Figure 5.2. Evaluation of the learned ROI embeddings using strength of 3HG's multi-hop connections (defined in Section 5.2.5). (a) The strength of 3HG's multi-hop connections calculated by the whole population with 1064 subjects; (b) The cosine similarity of each pair of the ROI embedding vectors learned by different multi-hop features (1-hop features, 2-hop features, and 3-hop features, respectively). For the matrices in both (a) and (b), the order of the brain regions is the same as the order defined in Destrieux atlas [1], where most of the first 44 regions are gyri while the last 31 regions are sulci.

strength of 3HG's multi-hop connection (Section 5.2.5) and displayed the results in Fig. 5.2. Figure. 5.2(a) shows the statistical results of the strength of 3HG's multi-hop connections based on the whole population of 1064 subjects. Figure. 5.2(b) shows the cosine similarity between the learned ROI embedding vectors. For each matrix, the order of the brain regions is the same as the order defined in Destrieux atlas [1], where most of the first 44 regions are gyri and the last 31 regions are sulci. For the ease of better analyzing the results, we divided all the 3HG's multi-hop connections into 3 different groups: gyri-gyri connections, gyri-sulci connections, and sulcisulci connections. The gyri-gyri connections mean both two connecting 3HGs are in gyral regions. The gyri-gyri connections are located at the top left of each matrix in Fig. 5.2. The gyri-sulci connections and sulci-sulci connections are defined in the same manner. In the matrices, the gyri-sulci connections are located at top right and bottom left, sulci-sulci connections are located at the bottom right. It is worth noting that the ground truth matrix (Fig. 5.2(a)) reflects the strength of actual anatomical relationship between two ROIs on cortical surface, e.g., the number of GyralNet edges between these two ROIs. While the embedding similarity matrix is defined based on cosine similarity between the embedding vectors of these two ROIs, which represents the relationship of these two ROIs in the latent (embedding) space. Therefore, if the two matrices have similar patterns, we can conclude that our ROI embeddings can effectively represent anatomical ROIs in the latent space including their relations. In Fig. 5.2, we can see that when only considering 1-hop features (the first column), the ground truth matrix shows strong gyri-gyri connections, weak gyri-sulci connections and almost no sulci-sulci connections. Our embeddings effectively capture this pattern, though some weak connections in the ground truth matrix are missing due to the sparsity of the input 1-hop features. When using 2-hop and 3-hop features (the second and third columns), the ground truth matrices show stronger connections in some regions (highlighted by red squares), and the same pattern can also be found in our embedding similarity matrices. Besides the visualization, we also adopted three measures, including structural similarity index measure (SSIM), Pearson correlation coefficient (PCC), and cosine similarity (CS), to quantitatively measure the similarity between the ground truth matrices and our embedding similarity matrices. The results are reported in Table 5.1. For all the three measures, the learned embed-

Table 5.1. Similarity between ground truth matrix and embedding similarity matrix.

Measures	**Multi-Hops**		
	1-hop	**2-hop**	**3-hop**
SSIM	*0.61*	*0.81*	*0.28*
PCC	*0.48*	*0.48*	*0.43*
CS	*0.47*	*0.49*	*0.43*

ding similarity matrices show highly consistent pattern to the ground truth matrices, especially for the 2-hop embedding whose SSIM measure is over 0.8. These results demonstrate that our learned embeddings can effectively encode the population-level common connection patterns in the data: if two ROIs have strong/weak connections on GyralNet (cortical space), they will also have large/small similarities in the latent space (embedding space).

5.3.3 Effectiveness of 3HGs Individual Embeddings

After the model is well trained by dataset-1, we applied the learned ROI embeddings and the model to the new dataset-2 to generate individual 3HG embeddings $-E^i_F$ (defined in (5.2)). According to the discussion in Section 5.3.2, the 2-hop features can provide the best embedding performance, hence we adopted 2-hop embedding in this section. As discussed in Section 5.2.5, the individual 3HG embeddings are expected to be able to preserve the individuality of 3HGs and infer reliable cross-subject anatomical correspondence. Therefore, in this section we will evaluate the effectiveness of the individual embeddings from two aspects.

5.3.3.1 Inferring Reliable 3HGs Cross-Subject Anatomical Correspondence

To evaluate the effectiveness of 3HG individual embeddings on inferring cross-subject anatomical correspondence, we used the 3HGs from one randomly selected subject (sub-0 in Fig. 5.3) as the exemplars to find their corresponding 3HGs in other subjects by the learned embedding vector $-E_F^i$. For each exemplar 3HG in sub-0, the correspondence inferring process is based on the following steps: 1) we examined all the 3HGs in different subjects and calculated the cosine similarity between the embedding vector of exemplar 3HG with each of the other 3HGs; 2) for each subject, the 3HGs that have the cosine similarity of 1.0 to the exemplar 3HG will be identified as the corresponding 3HGs in this subject; 3) if there is no 3HGs having the cosine similarity of 1.0, the one with the largest cosine similarity (above a threshold) will be identified as the corresponding 3HG. Following these steps, we obtained the corresponding 3HGs for each of the exemplar 3HGs in different subjects. In the sub-0,190 and 175 3HGs have been identified in the left and right hemisphere by the 3HGs identification pipeline (Section 5.2.2), respectively. For better visualization, we selected 60 3HGs on each hemisphere which spread over the whole cerebral cortex and showed the corresponding 3HGs on 10 randomly selected subjects in Fig. 5.3. Bubbles indicate the locations of 3HGs. The corresponding 3HGs in different subjects were color-coded by correspondence indices. From the results we can see that the corresponding 3HGs identified on different individuals have consistent locations in terms of common anatomical landscapes: for example, 3HG #157 in left hemisphere and 3HG #96 in right hemisphere (marked by red arrows) are found in the middle of left front superior gyri and middle of the right precentral gyri, respectively, across all the subjects.

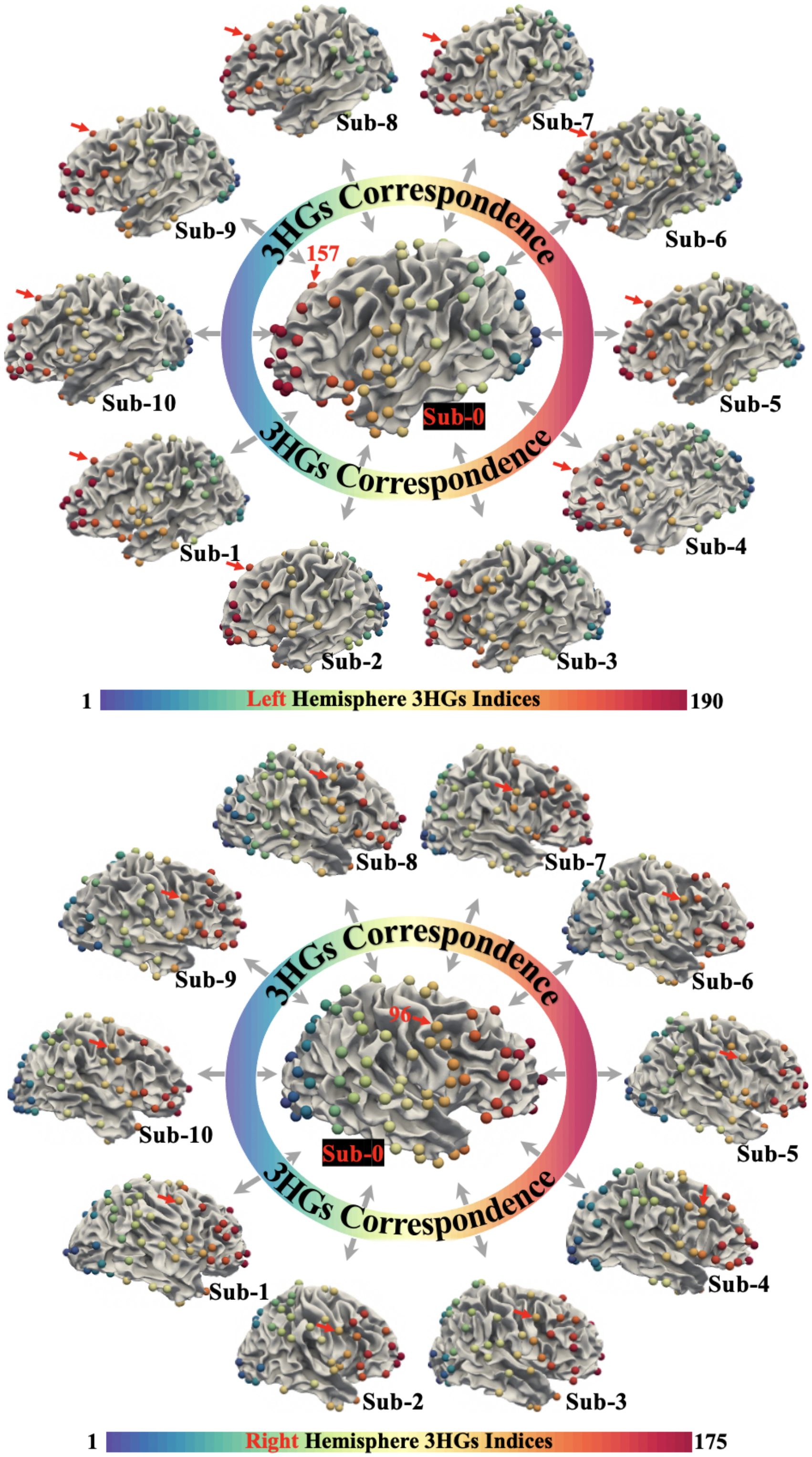


Figure 5.3. Cross-subject correspondences of 3HGs.

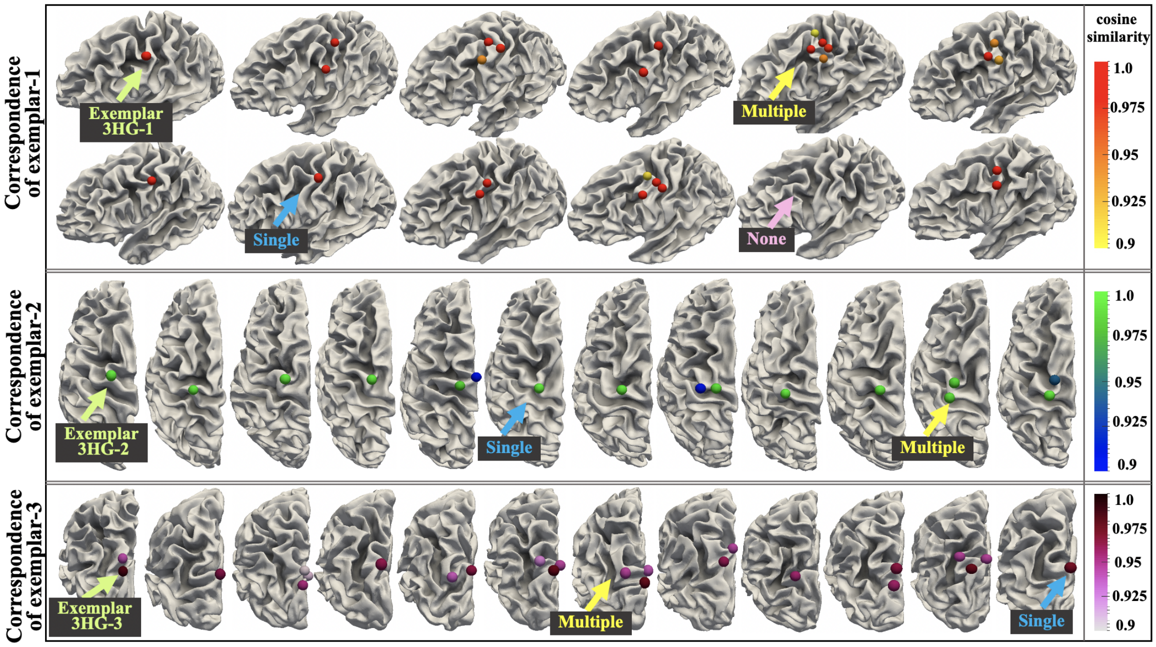


Figure 5.4. Cross-subject individuality and variability. We used three 3HGs (three blocks: top, middle, and bottom) as examples to show their correspondences in different subjects. We randomly selected 12 subjects and adopted the cosine similarity of 0.9 as the threshold. The corresponding 3HGs are represented by bubbles and the cosine similarity was encoded by the color of the bubbles. Due to the individuality, it is possible to find zero, one or multiple correspondences across different subjects.

5.3.3.2 Preserving Cross-subject Individuality

As an essential characteristic of human cerebral cortex, the folds of different subjects have shown intensive variability. To illustrate that the learned 3HG individual embeddings can preserve individuality of different subjects, we randomly selected three 3HGs as exemplars to find their corresponding 3HGs in different subjects. That is, within each subject, all the 3HGs that have a cosine similarity over 0.9 to the exemplar 3HG will be identified as the corresponding 3HG of that exemplar. We randomly selected 12 subjects and showed the results in Fig. 5.4. The cerebral cortex of the 12 subjects displays distinct cortical folding patterns. For example, the first exemplar 3HG is located at the conjunction of precentral gyri and front middle gyri, and there is no other 3HGs in the neighborhood. However, the folding patterns of the fifth subject (highlighted in yellow) is more convoluted, as a result, multiple 3HGs are aggregated at the same location. For the eleventh subject (highlighted in pink), there is no conjunction that can connect precentral gyri and front middle gyri, therefore, there is no 3HGs located at this location. More examples have been found and marked in exemplar 2 and 3. Despite the widely existed individuality of cortical folding patterns, our embedding method can provide a reliable way to identify the complex many-to-many correspondences without mapping different individual brains to the same space, and thus, the variabilities of 3HGs can be preserved. Notably, the embeddings and models used in this correspondence task were trained in a different dataset - dataset-1 by a self-supervised regression task, but they can generalize well on the new dataset-2 that shows promising transferable capability in other datasets. Therefore, the proposed framework can provide an effective way to design practical pre-training paradigms and facilitate downstream tasks in brain anatomy studies.

5.3.4 Regression Performance

The proposed framework was trained through a self-supervised regression task in a hierarchical twostage decoding manner. The first stage (Stage-1) reconstructs the hierarchical multi-hop embeddings from the combined embedding vector, whereas the second stage (Stage-2) recovers the multi-hop features from the hierarchical multi-hop embeddings. In this section, we used four metrics to evaluate the regression performance of the two decoding stages from various perspectives, including Mean Absolute Error (MAE) and Mean Squared Error (MSE) for magnitude, and Structural Similarity Index Measure (SSIM) and Cosine Similarity (CS) for overall pattern. In addition, we evaluated the regression performance of both the multi-hop embeddings/features and the embedding/feature vector of each single hop. The results

Table 5.2. Regression performance of 2-hop embedding.

Decoding Stage	**Hop**	**Metrics (mean±std)**			
		MAE ↓	**MSE ↓**	**SSIM ↑**	**CS ↑**
Stage-1	0th hop	0.007±0.004	0.001±0.001	0.910±0.067	0.900±0.135
	1th hop	0.014±0.007	0.004±0.003	0.941±0.034	0.993±0.011
	2th hop	0.043±0.020	0.038±0.018	0.969±0.010	0.999±0.001
	All	0.021±0.009	0.014±0.007	0.972±0.008	0.997±0.003
Stage-2	0th hop	0.022±0.005	0.005±0.004	0.622±0.063	0.792±0.231
	1th hop	0.033±0.012	0.018±0.017	0.645±0.082	0.934±0.104
	2th hop	0.079±0.031	0.150±0.163	0.632±0.105	0.961±0.077
	All	0.045±0.014	0.058±0.060	0.669±0.085	0.956±0.080

***Stage-1**: Reconstruction of hierarchical multi-hop embeddings –* $\tilde{E}^{i}_{MH}$.

***Stage-2:** Reconstruction of the input multi-hop features –* $\tilde{F}^{i}_{MH}$ *(for more information, see eq-(1) and Fig. 1).*

l^{th} ***hop:** considers the single hop; **All:** considers multiple hops covering* 0th, 1th *and* 2th *hops.*

↓: *a smaller value indicates better performance;* **↑:** *a larger value indicates better performance.*

were calculated using 169,923 3HGs in dataset-2 (independent testing dataset) and reported in Table 5.2. Our results show that the reconstructed embeddings/features in the first/second decoding stage have low MAE and MSE (¡0.1/0.15) and high SSIM and CS (¿0.9/0.6), indicating that the two-stage decoding framework performs well in the regression task. Furthermore, it is worth noting that the performance of stage-1 is slightly better than stage-2, with lower MAE and MSE, and higher SSIM and CS. This may be because the ground truths in stage- 2 are highly sparse matrices - the multi-hop features created by one-hot encoding, whereas the ground truths in stage-1 are dense embedding vectors. In the discussion section, we will compare the one-hot encoding and the learnable embedding vectors further.

5.4 Discussion

Self-Supervised Embedding. A common problem in deep learning models is limited samples: the huge architectures usually demand hundreds of millions of labeled data which are often publicly inaccessible. Especially in neural imaging domains, where designing these labeled data can be a time-consuming and expensive process and impossible in some scenarios. In natural language processing (NLP), this appetite for data has been successfully addressed by self-supervised pretraining, which enables training of generalizable NLP framework containing over one hundred billion parameters, such as BERT [179] and GPT [187, 188, 189]. Inspired by these successful models in NLP, in this study, we adopted an autoencoder architecture – a simple self-supervised method, to design the learning-based embedding framework. In our experiments, the proposed framework generalizes well on new datasets and shows promising transferable capability in downstream tasks.

Disentangling the commonality and individuality. Our proposed embedding framework was trained to indirectly encode brain anatomy using folding pattern derived landmarks – 3HGs. Different from NLP methods in which the words are well-defined, and the language vocabulary can be easily built up, there is no pre-exist "vocabulary" when representing cerebral cortex, since each brain has unique folding patterns. To solve this problem, we designed a new embedding strategy to disentangle the commonality and individuality of the 3HGs: instead of embedding the 3HG itself, we embedded the ROIs (from brain atlas) into a set of ROI embedding vectors, serving as basic blocks for representing commonality, and then used these ROI embedding vectors to distill individuality.

Integrating multi-modal data. In this work, we limited our interest in the folding patterns of 3HGs and focused on the effectiveness of the proposed method on anatomical correspondences. We didn't include the white matter structure into our study. Although it has been widely reported that brain folding patterns are closely related to brain structural connectivity patterns [164, 165, 166], there is still no consensus about the relationship between them. For example, some studies suggested that the tension on the axon pulls the cortex closer and forms the gyri [190, 163], while in other works gyri were reported to be connected by axons with greater density than sulci at different scales [191, 192]. There are also some studies suggesting that there exists a superficial axonal system at the boarder of white matters and gray matters which could impede the detection of axonal connections, especially in sulci regions [193]. In addition, the disease related alterations of white matter structures make the situation even more complicated [165, 166, 94, 167, 168, 169]. However, it is undeniable that white matter plays an important role in the formation of folding patterns, and we intend to include it in our future studies: 1) Investigating whether the corresponding 3HGs with similar anatomical characteristics also have similar fiber connection patterns; 2) Identifying a group of 3HGs with similar anatomical features and fiber connection patterns and investigating their functional homogeneity/heterogeneity; 3) Incorporating fiber connection patterns and function into current frameworks to establish a more comprehensive 3HGs map with superb functional homogeneity and intrinsically established cross-subject cortical correspondences. Then, based on this map of 3HGs, we can radiate the research scope to larger area and to include more landscapes, such as 2-hinges.

One-hot encoding vs learned embeddings. The ROI features of 3HGs were initially represented by one-hot encodings and used as input features to learn anatomical meaningful embeddings. Compared to the learned embeddings, one-hot vectors cannot be directly used as an embedding vector to infer cross-subject correspondence for two reasons: 1) One-hot vectors are anatomical meaningless and cannot provide reliable cross-subject 3HGs correspondences. Each one-hot vector contains a single one and $\mathrm{N}-1$ zeros where N is the number of dimensions. As a result, ROIs are embedded in isolation and are equal distance apart, making it impossible to represent the underlying relationships between ROIs. In this way, the similarity of two 3HGs based on one-hot encoding is only related to the number of common ROIs shared by their multi-hop features, while the underlying connections between ROIs are ignored, rendering it powerless when inferring correspondences of 3HGs. For example, under one-hot encoding, the anatomical similarities of all 3HG pairs with no common ROIs of their multi-hop features are zero. In contrast, the learned embeddings can effectively encode the population-level common connection patterns between ROIs, with closely connected ROIs having similar embedding vectors, and thus can provide more reliable anatomical correspondences; 2) One-hot vectors are sparse and grow with vocabulary size, which can easily lead to the curse of dimensionality, whereas embeddings are dense and low-dimensional, making them more computationally efficient. In general, the learned embeddings are more efficient than the one-hot vectors in both anatomical correspondence reliability and computational efficiency.

5.5 Conclusion

In this work, we proposed a learning-based embedding framework to embed the anatomically meaningful patterns of 3HGs, into a group of learnable embedding vectors. Each 3HG can be represented as a hierarchical combination of the learned embedding vectors via a set of multi-hop combination coefficients. By this way, the regularity of folding pattern is encoded into the embedding vectors, and the variability is preserved by the individual-specific coefficients. We evaluated the proposed method using HCP dataset and the experiment results show that the learned embeddings can successfully encode the cortical folding patterns and reliably infer the cross-subject complex many-tomany correspondences of 3HGs.

CHAPTER 6

LARGE LANGUAGE MODELS (LLMS) IN HEALTHCARE DOMIAN

Large language models (LLMs) – the latest generation of AI models—are large-scale machine learning models trained on massive, diverse datasets and can be applied to numerous downstream tasks. However, LLMs are primarily pretrained on general-domain datasets, their performance in specialized fields such as healthcare remains incompletely evaluated. To address this, in **Chapter 6**, we conducted extensive assessments across various real-world tasks in healthcare domain. These evaluations allowed us to scrutinize a key question: whether model designs should be generic or domain specific. In addition, with the goal of unlocking the potential benefits of LLMs in improving healthcare outcomes, we have developed effective approaches to tailor these models to the unique requirements of specialized domains.

6.1 Evaluation of LLMs in Clinical NLP

The rise of LLMs has marked a pivotal shift in the field of natural language processing (NLP) and also have made a significant impact in the medical field. However, a comprehensive evaluation of the performance of these models in medical domain remains to be conducted. This lack of assessment is especially apparent within the context of radiology NLP. The research work in this section seeks to bridge this gap by critically evaluating thirty two LLMs in interpreting radiology reports, a crucial component of radiology NLP. Specifically, the ability to derive impressions from radiologic findings is assessed. The outcomes of this evaluation provide key insights into the performance, strengths, and weaknesses of LLMs, informing their practical applications within the medical domain.

6.1.1 Datasets

Our study utilizes two comprehensive and publicly available datasets, the MIMIC-CXR and the OpenI datasets. These datasets were utilized to test the performance and efficacy of various LLMs in generating radiology text reports. In our study, we used these datasets to evaluate the capabilities of the LLMs. We focused on the "Findings" and "Impression" sections of each report as they provide comprehensive and detailed textual information about the imaging findings and the radiologists' interpretations.

1. **MIMIC-CXR Dataset** The MIMIC-CXR dataset is a substantial repository of de-identified chest radiographs (CXRs) that are complemented with their corresponding radiology reports. The dataset contains medical data from over 60,000 patients who were admitted to the Beth Israel Deaconess Medical Center between 2001 and 2012. The radiology reports in the MIMIC-CXR dataset typically consist of two sections: "Findings" and "Impression". The "Findings" section details observations from radiology images, while the "Impression" section provides summarized interpretations of these observations.
2. **OpenI Dataset.** The OpenI dataset is another essential resource that was used in our study. It is a freely available repository that consists of radiology images paired with their respective reports. This dataset provided an independent external platform to validate the performance and generalizability of our LLMs across different data sources.

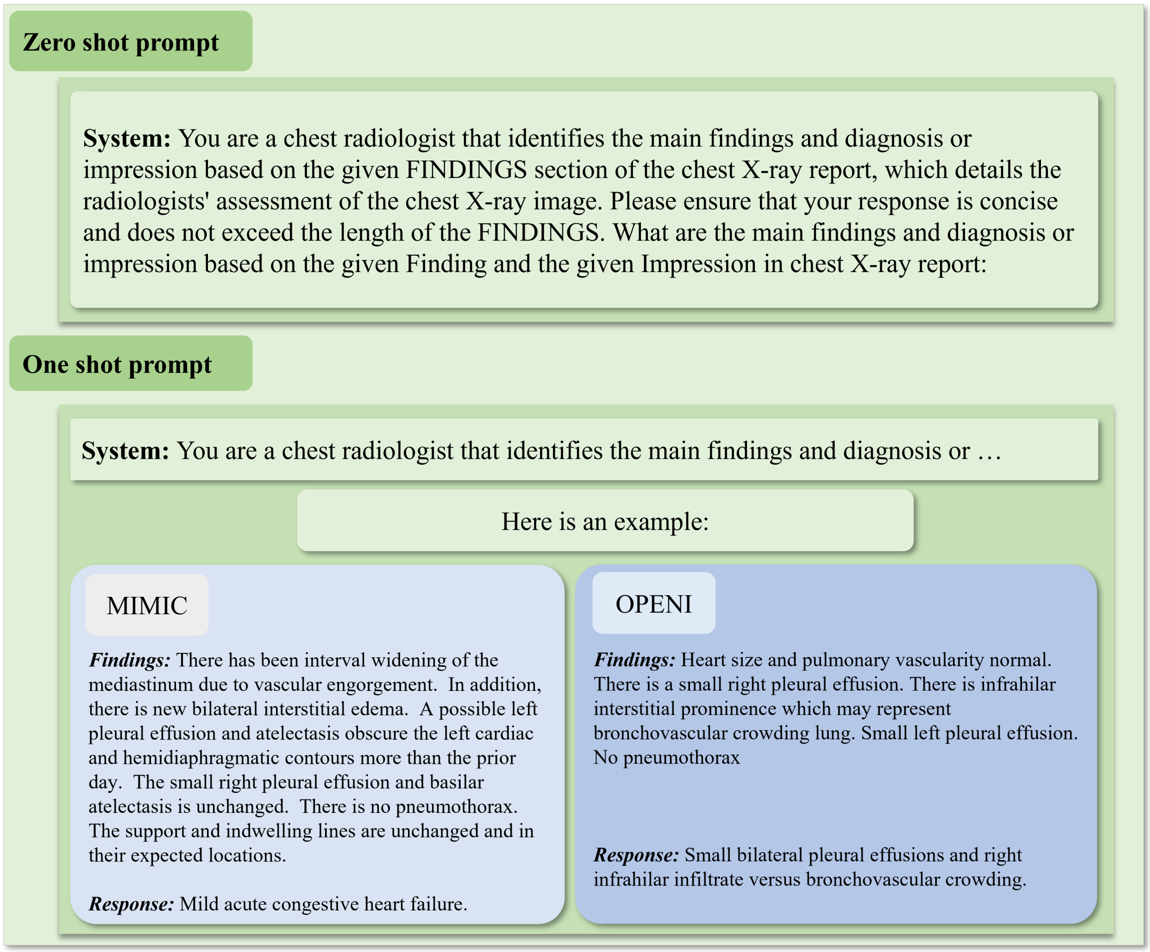
We followed an existing literature approach to randomly divide the dataset into separate segments for testing purposes. This division resulted in a subset of 2400, 292, and 576 reports for various testing scenarios.

6.1.2 Methods

This section will discuss our testing methods for LLMs. We will begin by introducing the datasets MIMIC and OpenI, which we use for evaluation. Our testing approach involves employing a fixed set of prompts and parameters to assess the performance of LLMs in the field of radiology, specifically focusing on deriving impression-based performance from findings. To ensure consistency, we set several hyperparameters of the LLMs, namely the temperature to 0.9, the top_k to 40, and the top_p to 0.9. To evaluate the model's zero-shot and few-shot performance, we utilize zero-shot, one-shot, and five-shot examples as prompts. The experimental results and their detailed analysis are presented in the results section.

6.1.2.1 Testing Approach

Our testing approach involves utilizing a fixed set of prompts and parameters to evaluate the LLMs. The model's inference parameters, namely the temperature, top_k, and top_p, are fixed at 0.9, 40, and 0.9, respectively, to ensure consistency. We engage zero-shot, one-shot, and five-shot prompts to examine the model's zero-shot and few-shot performance. A zero-shot prompt involves presenting the model with a new task, with no prior examples provided. A one-shot prompt involves providing the model with one prior example, while a five-shot prompt provides the model with five prior examples. This variation in prompts offers a nuanced understanding of how the LLMs operate under different conditions and degrees of prior exposure.



Zero shot prompt

System: You are a chest radiologist that identifies the main findings and diagnosis or impression based on the given FINDINGS section of the chest X-ray report, which details the radiologists' assessment of the chest X-ray image. Please ensure that your response is concise and does not exceed the length of the FINDINGS. What are the main findings and diagnosis or impression based on the given Finding and the given Impression in chest X-ray report:

One shot prompt

System: You are a chest radiologist that identifies the main findings and diagnosis or …

Here is an example:

MIMIC

Findings: There has been interval widening of the mediastinum due to vascular engorgement. In addition, there is new bilateral interstitial edema. A possible left pleural effusion and atelectasis obscure the left cardiac and hemidiaphragmatic contours more than the prior day. The small right pleural effusion and basilar atelectasis is unchanged. There is no pneumothorax. The support and indwelling lines are unchanged and in their expected locations.

Response: Mild acute congestive heart failure.

OPENI

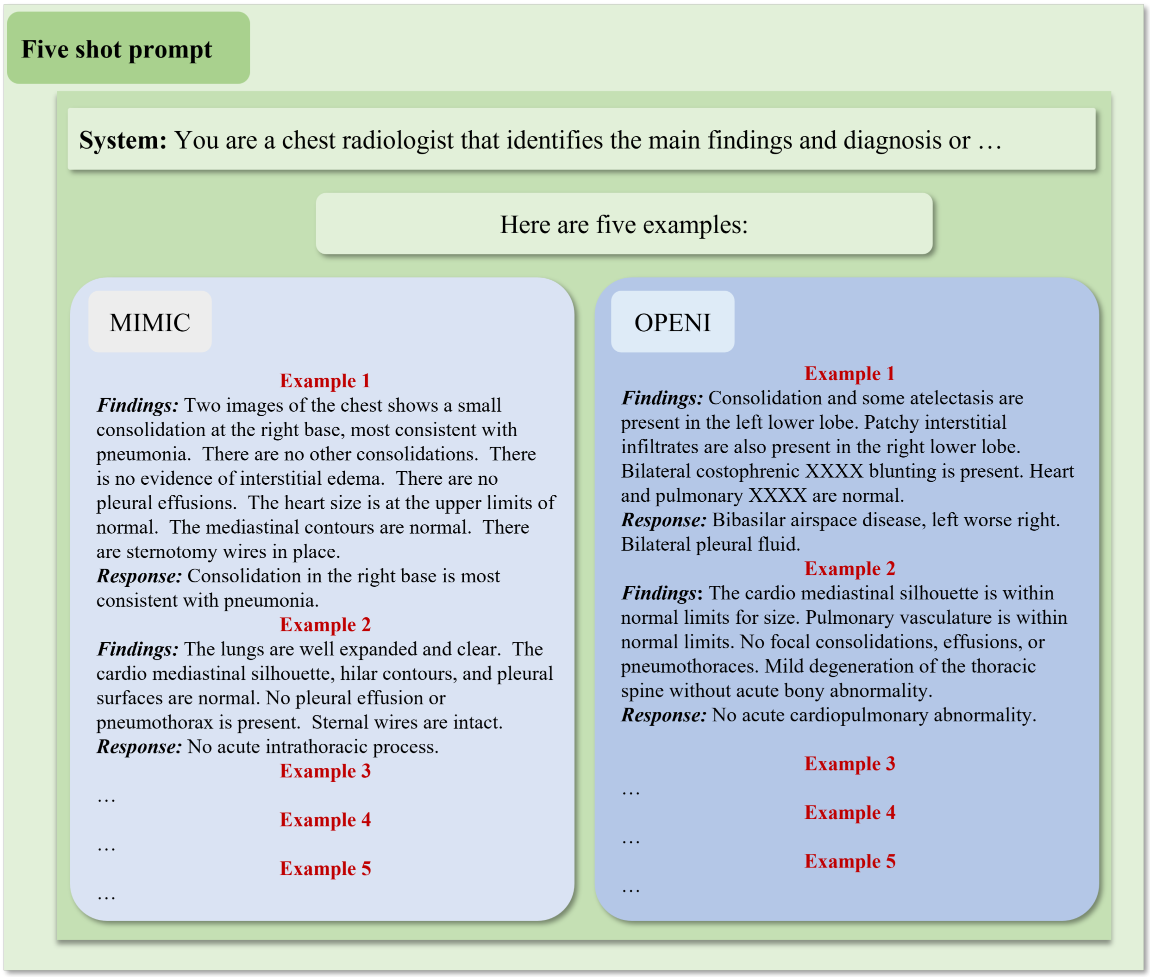
Findings: Heart size and pulmonary vascularity normal. There is a small right pleural effusion. There is infrahilar interstitial prominence which may represent bronchovascular crowding lung. Small left pleural effusion. No pneumothorax

Response: Small bilateral pleural effusions and right infrahilar infiltrate versus bronchovascular crowding.

Figure 6.1. Zero shot prompt and one shot prompt used in the experiment..

6.1.2.2 Model Selection

Considering both resource constraints and the need for uniformity in model comparison, our evaluation specifically focuses on Large Language Models (LLMs) with approximately 7 billion parameters. The choice of this parameter count is based on two primary considerations. First, models of this size strike a balance between computational efficiency and model performance. They allow for faster inference, making it feasible to thoroughly evaluate the models over the complete testing dataset in a practical timeframe. Second, this parameter count is well-represented across



Five shot prompt

System: You are a chest radiologist that identifies the main findings and diagnosis or …

Here are five examples:

MIMIC

Example 1

Findings: Two images of the chest shows a small consolidation at the right base, most consistent with pneumonia. There are no other consolidations. There is no evidence of interstitial edema. There are no pleural effusions. The heart size is at the upper limits of normal. The mediastinal contours are normal. There are sternotomy wires in place.

Response: Consolidation in the right base is most consistent with pneumonia.

Example 2

Findings: The lungs are well expanded and clear. The cardio mediastinal silhouette, hilar contours, and pleural surfaces are normal. No pleural effusion or pneumothorax is present. Sternal wires are intact.

Response: No acute intrathoracic process.

Example 3

…

Example 4

…

Example 5

…

OPENI

Example 1

Findings: Consolidation and some atelectasis are present in the left lower lobe. Patchy interstitial infiltrates are also present in the right lower lobe. Bilateral costophrenic XXXX blunting is present. Heart and pulmonary XXXX are normal.

Response: Bibasilar airspace disease, left worse right. Bilateral pleural fluid.

Example 2

***Findings*:** The cardio mediastinal silhouette is within normal limits for size. Pulmonary vasculature is within normal limits. No focal consolidations, effusions, or pneumothoraces. Mild degeneration of the thoracic spine without acute bony abnormality.

Response: No acute cardiopulmonary abnormality.

Example 3

…

Example 4

…

Example 5

…

Figure 6.2. Five shot prompt used in the experiment..

different types of LLMs, allowing for a broad and diverse range of models to be included in the study.

For open-source models, we procure the necessary code and model parameters directly from their official GitHub repositories. These repositories provide comprehensive documentation and community support, ensuring that the models are implemented and evaluated correctly.

For commercially available models, such as Sensenova, ChatGPT, GPT-4, PaLM2, and Anthropic Claude2, we utilize their respective Application Programming Interfaces (APIs). These APIs offer a structured and standardized way of interacting with the models, enabling us to input our pre-determined prompts and parameters and receive the model outputs in a consistent and reliable manner. The model summary is as follows:

1. **HuatuoGPT** is a language model developed by the Shenzhen Research Institute of Big Data from the Chinese University of Hong Kong, Shenzhen. HuatuoGPT-7B is trained on the Baichuan-7B corpus, while HuatuoGPT-13B is based on Ziya-LLaMA-13B-Pretrain-v1. The advantage of HuatuoGPT is in its integration of real-world medical data and the information-rich base of ChatGPT. This allows HuatuoGPT to provide detailed diagnoses and advice in medical consultation scenarios, similar to a doctor's approach [194]. HuatuoGPT has two versions: HuatuoGPT-7B and HuatuoGPT-13B. In our experiments, we used the HuatuoGPT-7B version.
2. **Luotuo** is a Chinese language model exploited and maintained by the researchers Qiyuan Chen, Lulu Li, and Zihang Leng. Luotuo is fine-tuned by the LLaMA on Chinese corpus utilizing LoRA technique and does well in Chinese infering [195]. Luotuo has three versions: Luotuo-lora-7b-0.1, Luotuo-lora-7b-0.3, and luotuo-lora-7b-0.9. Luotuo-lora-7b-0.3 was used in the experiments.
3. **Ziya-LLaMA** [196] denotes bilingual pre-trained language models based on LLaMA. It is a member of the open-source general large model series and is introduced by the Center for Cognitive Computing and Natural Language Research (CCNL) at the IDEA Research Institute. Ziya-LLaMA boasts remarkable versatility, demonstrating proficiency across a wide array of tasks including translation, programming, text classification, information extraction,

summarization, copywriting, common sense Q&A, and mathematical calculation. Its comprehensive training process comprises three stages: large-scale continual pre-training, multi-task supervised fine-tuning, and human feedback learning. Ziya has four version, Ziya-LLaMA-13B-v1.1, Ziya-LLaMA-13B-v1, Ziya-LLaMA-7B-Reward, and Ziya-LLaMA-13B-Pretrain-v1. In this study, we investigated the Ziya-LLaMA-13B-v1.

4. **YuYan-Dialogue** YuYan-Dialogue [197] is a Chinese language dialogue model by fine-tuning the YuYan-11b on a large multi-turn dialogue dataset of high quality and developed by Fuxi AI lab, Netease.Inc. It is trained on a large Chinese novel dataset of high quality and has very strong conversation generation capabilities. YuYan-Dialogue has only one version that is YuYan-Dialogue. Therefore, we used it in our experiments.
5. **BenTsao** BenTsao [198] is a medical language model based on LLaMA-7B model developed by SCIR Lab in Harbin Institution of Technology. It has undergone Chinese medical instruction fine-tuning and instruction tuning. They built a Chinese medical instruction dataset through the medical Knowledge graph and GPT3.5 API, based on which, they further fine-tuned the model, improving the question-and-answer effect of LLaMA in the medical field. BenTsao has four versions, LLaMA-med, LLaMA-literature, Alpaca-med, Alpaca-all-data. Here, we used the LLaMA-med (BenTsao) for comparison.
6. **XrayGLM** Xray-GLM [199] is a vision-language model developed by Macao Polytechnic University. It is based on the VisualGLM-6B and fintuned on the translated Chinese version MIMIC-CXR, OpenI dataset. It has strong ability on chest Xray VQA. Here, we used the newest version of the Xray-GLM for comparison.

7. **ChatGLM-Med** ChatGLM-Med [200] is a language model developed by SCIR Lab in Harbin Institution of Technology. It is based on the ChatGLM-6b and has undergone Chinese medical instruction fine-tuning and instruction tuning. They built a Chinese medical instruction dataset through the medical Knowledge graph and GPT3.5 API, and on this basis, and fine-tuned the model based on the instructions of ChatGLM-6B, improving the question-and-answer effect of ChatGLM in the medical field. Here, we chose the newest version of ChatGLM-Med model for comparison.
8. **ChatGPT/GPT4** ChatGPT and GPT4 are both highly influential large language models developed by OpenAI. The full name of ChatGPT is gpt-3.5-turbo, which is developed on the basis of gpt2 and gpt3.The training process of ChatGPT mainly refers to instructGPT [201], ChatGPT is an improved instructionGPT. The main difference from GPT-3 [189]. is that the new addition is called RLHF (Reinforcement Learning from Human Feedback, human feedback reinforcement learning) [202]. This training paradigm enhances human conditioning of the model output and enables a more comprehensible ranking of the results. ChatGPT has strong language understanding ability and can handle various language expressions and queries. ChatGPT has an extensive knowledge base that can answer various frequently asked questions and provide useful information. GPT-4 is a successor to GPT-3, so it may be more capable in some ways. In our experiments, we used the ChatGPT and GPT4.
9. **ChatGLM2/ChatGLM** ChatGLM2 is a large language model developed by Tsinghua University, developed on the basis of the ChatGLM using the GLM framework [203]. ChatGLM2 has more powerful performance, which can handle longer contexts and perform more efficient reasoning with a more open protocol. What's more, ChatGLM2 is an excellent bilingual pre-trained model [204].

There are many versions of ChatGLM2 depending on the size of the pattern instruction set. This work mainly tested ChatGLM2-6B and ChatGLM-6B.

10. **QiZhenGPT** QiZhenGPT [205] is a model developed by Zhejiang University. It uses the Chinese medical instruction data set constructed by QiZhen Medical Knowledge Base, and based on this, performs instruction fine-tuning on the Chinese-LLaMA-Plus-7B, CaMA-13B, and ChatGLM-6B models. QiZhenGPT has an excellent effect in Chinese medical scenarios, and it is more accurate in answering questions than ChatGLM-6B. According to different model objects fine-tuned by instructions, QizhenGPT has three types [206]: QiZhen-Chinese-LLaMA-7B, QiZhen-ChatGLM-6B, and QiZhen-CaMA-13B. In this work, we tested mainly on QiZhen-Chinese-LLaMA-7B.
11. **MOSS** MOSS-MOON-003 is the third version of the open-sourced plugin-augmented bilingual (i.e. Chinese and English) conversational language model MOSS, specifically from the MOSS-MOON-001 to MOSS-MOON-003, developed by the OpenLMLab from Fudan University [207]. The MOSS-MOON-003-sft is fine-tuned with supervision on approximately 1.1M multi-turn conversational data to the base model, MOSS-MOON-003-base. The advantage of MOSS-MOON-003 is it can follow bilingual multi-turn dialogues, refuse inappropriate requests and utilize different plugins due to its base model (i.e. MOSS-MOON-003-base was pre-trained on 700B English, Chinese, and code tokens), fine-tuning on multi-turn plugin-augmented conversational data, and further preference-aware training. There are 10 versions available: MOSS-MOON-003-base, MOSS-MOON-003-sft, MOSS-MOON-003-sft-plugin, MOSS-MOON-003-sft-int4, MOSS-MOON-003-sft-int8, MOSS-MOON-003-sft-plugin-int4, MOSS-MOON-003-sft-plugin-int8, MOSS-MOON-003-pm, MOSS-MOON-003, and MOSS-

MOON-003-plugin. In our experiments, we used the MOSS-MOON-003-sft version.

12. **ChatFlow** ChatFlow [208] is a fully-parameterized training model developed by the Linly project team, built upon the foundations of LLaMa and Falcon and based on the TencentPretrain pre-training framework [209] and a large-scale Chinese scientific literature dataset [210]. By utilizing both Chinese and Chinese-English parallel incremental pre-training, it transfers its language capabilities from English to Chinese. The key advantage of ChatFLow is that it addresses the issue of weaker Chinese language understanding and generation abilities found in the open-source models Falcon and LLaMa. It significantly improves the encoding and generation efficiency of Chinese texts. ChatFlow comes in two versions, namely ChatFlow-7B and ChatFlow-13B. For our experiments, we utilized the ChatFlow-7B version.
13. **CPM-Bee** CPM-Bee [211] is a large model system ecology based on OpenBMB, and it is a self-developed model of the Facing Wall team. It is a completely open source, commercially available Chinese-English bilingual basic model, and it is also the second milestone achieved through the CPM-Live training process. CPM-Bee uses the Transformer autoregressive architecture, with a parameter capacity of tens of billions, pre-training on a massive corpus of trillions of tokens, and has excellent basic capabilities. There are four versions of CPM-Bee: CPM-Bee-1B, CPM-Bee-2B, CPM-Bee-5B, CPM-Bee-10B. In this experiment, we tested the performance of CPM-Bee-5B (CPM-Bee).
14. **PULSE** The PULSE model [212] is a large-scale language model developed on the OpenMEDLab platform. It is based on the OpenChina LLaMA 13B model, which is further fine-tuned using approximately 4,000,000 SFT data from the medical and general domains. PULSE supports a variety of natural language

processing tasks in the medical field, including health education, physician exam questions, report interpretation, medical record structuring, and simulated diagnosis and treatment. PULSE has two versions, PULSE_7b and PULSE_14b. In this experiment, we tested the version of PULSE_7b.

15. **Baichuan** Baichuan, developed by Baichuan Intelligence, is a large pre-trained model based on the Transformer architecture. The baichuan-7B model, comprising 7 billion parameters, was trained on approximately 12 trillion tokens, utilizing the same model design as LLaMa. Subsequently, they further developed the baichuan-13B model, which is even larger in size and trained on a greater amount of data [213]. The key advantage of the Baichuan model lies in its use of an automated learning-based data weighting strategy to adjust the data distribution during training, resulting in a language model that supports both Chinese and English. It has demonstrated robust language capabilities and logical reasoning skills across various datasets. Two versions of the Baichuan model are developed: baichuan-7B and baichuan-13B. For our experiments, we utilized the baichuan-7B version.
16. **AtomGPT** AtomGPT [214], developed by Atom Echo, is a large language model based on the model architecture of LLaMA [215]. AtomGPT uses a large amount of Chinese and English data and codes for training, including a large number of public and non-public data sets. Developers use this method to improve model performance. AtomGPT currently has four versions: AtomGPT_8k, AtomGPT_14k, AtomGPT_28k, AtomGPT_56k. In this experiment, we chose AtomGPT_8k for testing.
17. **ChatYuan** ChatYuan [216] large v2 is an open-source large language model for dialogue, supports both Chinese and English languages, and in ChatGPT style. It is published by ClueAI. ChatYuan large v2 can achieve high-quality results

on simple devices that allows users to operate on consumer graphics cards, PCs, and even cell phones. It got optimized for fine-tuning data, human feedback reinforcement learning, and thought chain. Also, comparing with its previous version, the model is optimized in many language abilities, like better at both Chinese and English, generating codes and so on. ChatYuan has three versions: ChatYuan-7B, ChatYuan-large-v1, ChatYuan-large-v2. In our experiments, we tested the ChatYuan-large-v2.

18. **Bianque-2.0** Bianque [217] is a large model of healthcare conversations fine-tuned by a combination of directives and multiple rounds of questioning conversations. Based on BianQueCorpus, South China University of Technology chose ChatGLM-6B as the initialization model and obtained BianQue after the instruction fine-tuning training. BianQue-2.0 expands the data such as drug instruction instruction, medical encyclopedic knowledge instruction, and ChatGPT distillation instruction, which strengthens the model's suggestion and knowledge query ability. By using Chain of Questioning, the model can relate more closely to life and to improve questioning skills, which is different from most language model. It has two versions: Bianque-1.0 and Bianque-2.0. In our experiments, we tested Bianque-2.0.
19. **AquilaChat** AquilaChat [218] is a language model developed by the Beijing Academy of Artificial Intelligence. AquilaChat is an SFT model based on Aquila for fine tuning and Reinforcement learning. The AquilaChat dialogue model supports smooth text dialogue and multiple language class generation tasks. By defining extensible special instruction specifications, AquilaChat can call other models and tools, and is easy to expand its functions. AquilaChat has two versions: AquilaChat-7B and AquilaChat-33B. In our experiments, we used the AquilaChat-7B version.

20. **Aquila** Aquila [219] is a language model developed by the Beijing Academy of Artificial Intelligence. Aquila-7B is a basic model with 7 billion parameters. The Aquila basic model inherits the architectural design advantages of GPT-3, LLaMA, etc. in terms of technology, replaces a batch of more efficient low-level operator implementations, redesigns and implements the Chinese English bilingual tokenizer, upgrades the BMTrain parallel training method, and achieves nearly 8 times the training efficiency compared to Magtron+DeepSpeed Zero-2. Aquila has two versions: Aquila-7B and Aquila-33B. In our experiments, we used the Aquila -7B version.
21. **Chinese-Alpaca-Plus** Chinese-Alpaca-Plus [206] is a language model developed by Yiming Cui etc. Chinese-Alpaca-Plus is a language model based on LLaMA. Chinese-Alpaca-Plus has improved its coding efficiency and semantic understanding of Chinese by adding 20000 Chinese tags to the existing Glossary of LLaMA [206]. Chinese-Alpaca-Plus has three versions: Chinese-Alpaca-Plus-7B, Chinese-Alpaca-Plus-13B, and Chinese-Alpaca-Plus-33B. In our experiments, we used the Chinese-Alpaca-Plus-7B version.
22. **TigerBot** Tigerbot-7b-sft-v1 [220] is a language model developed by the Tigerbot Company. TigerBot-7b-sft-v1 is a large-scale language model with multiple languages and tasks. Tigerbot-7b-sft-v1 is an MVP version that has undergone 3 months of closed development and over 3000 experimental iterations.Functionally, Tigerbot-7b-sft-v1 already includes the ability to generate and understand most of the classes, specifically including several major parts: content generation, image generation, open-ended Q&A, and long text interpretation.Tigerbot-7b-sft has two versions: Tigerbot-7b-sft-v1 and Tigerbot-7b-sft-v2. In our experiments, we used the tigerbot-7b-sft-v1 version.

23. **XrayPULSE** XrayPULSE [221] is an extension of PULSE and made by OpenMEDLab. OpenMEDLab utilize MedCLIP as visual encoder and Q-former (BLIP2) following a simple linear transformation as the adapter to inject the image to PULSE. For aligning the frozen visual encoder and the LLM by the adapter, OpenMEDLab generate Chinese-version Xray-Report paired data from radiology. By extending PULSE, XrayPULSE is fine-tuned on Chinese-version Xray-Report paired datasets and aims to work as a biomedical multi-modal conversational assistant. The basic model is PULSE and we did the tests on XrayPULSE by modifying the Checkpoint file.
24. **DoctorGLM** DoctorGLM [222] is the first chinese diagnosis large language model (released at 3rd april 2023) that developed by ShanghaiTech University [222]. It is fine-tuned on ChatGLM-6B using real-world online diagnosis dialogue. DoctorGLM has several updates and two different parameter-efficient finetune setting (p-tuning and LoRA). In our experiments, we used the DoctorGLM-5-22 p-tuning version.
25. **Robin-7B-medical** Robin-medical (LMFlow) [223] is a toolkit providing a complete fine-tuning workflow for a large foundation model to support personalized training with limited computing resources. It is developed by Diao et al. from the Hong Kong University of Science and Technology. They provide a series of LoRA models based on the LLama model called Robin-medical, which are specially fine-tuned on the PubMedQA and MedMCQA datasets. The advantage of LMFlow is that it introduces an extensible and lightweight toolkit to simplify the fine-tuning and inference of general large foundation models. This allows people to fine-tune foundation models to mitigate the current status that most existing models exhibit a major deficiency in specialized-task applications.

Robin-medical has 7B, 13B, 33B and 65B versions. We tested the 7B version in our experiments.

26. **PaLM2** PaLM2 is a large language model developed by Google. PaLM 2 is a language model based on a tree structure, which makes use of the context and grammatical rules in the language to make the model's understanding of text information more refined, accurate and comprehensive. Different from traditional sequence-based models (such as GPT), PaLM2 uses some new methods that are more popular than traditional methods, such as Tree-LSTM [224], Bert [179], etc. Compare to PaLM, PaLM2 excels at advanced reasoning tasks including code and math, classification and question answering, translation and multilingualism It excels at advanced reasoning tasks including code and math, classification and question answering, translation and multilingualism. It's also being used in other state-of-the-art models, like Med-PaLM2 and Sec-PaLM. We tested the PaLM2 version in our experiments.
27. **SenseNova** SenseNova [225] is a large language model developed by SenseTime. Through the trinity flywheel of data, model training and deployment, it can provide various large models and capabilities such as natural language, content generation, automatic data annotation, and custom model training. Based on the previous accumulation of NLP work by SenseTime, SenseNova is still good in the domestic large language model. Based on the "SenseNova" large-scale model system, SenseTime has also developed a series of generative AI models and applications including Miahua SenseMirage, Ronin SenseAvatar, Qiongyu SenseSpace, and Gewu SenseThings. We mainly tested SenseNova in this work.
28. **Anthropic Claude2** Claude2 is a large language model developed by Anthropic, which is characterized by helpful and trustworthy. It is developed on the basis of Claude1.3. Anthropic uses a technical framework they call Consti-

tute AI [226] to achieve harmless processing of language models. Claude2 has a more powerful text processing function than GPT4, can handle larger-scale text, and has stronger context understanding ability and Chinese understanding ability. Claude is currently available in two versions, the powerful Claude, which excels at a wide range of tasks from complex dialogue and creative content generation to detailed instruction following, and the faster and more affordable Claude Instant, which also Can handle casual conversations, text analysis, summarization, and document question answering. We tested the latest version of Anthropic Claude2 for this work.

29. **BayLing** Bayling [227] is an instruction-following large language model equipped with advanced language alignment. It is a product from Natural Language Processing Group, Institute of Computing Technology, Chinese Academy of Science. BayLing can be effortlessly deployed on a consumer-grade GPU. It shows superior capability in English/Chinese generation, instruction following and multi-turn interaction. Bayling has three versions: BayLing-7B-v1.0, BayLing-13B-v1.0, BayLing-13B-v1.1. In our experiments, we tested BayLing-7B.

6.1.2.3 Uniform Testing Prompts

For a fair and equitable comparison across different LLMs, we adopt a uniform approach in the selection and use of testing prompts. The same prompts are used across all models and conditions, regardless of whether they are zero-shot, one-shot, or five-shot scenarios.

In a zero-shot evaluation, the models are presented with a new task, with no prior examples given. For the one-shot scenario, we provide the model with one prior example. Meanwhile, in the five-shot scenario, the model is given five examples to learn from. These scenarios aim to mimic real-world usage conditions where models are given a limited number of examples and are expected to generalize from them.

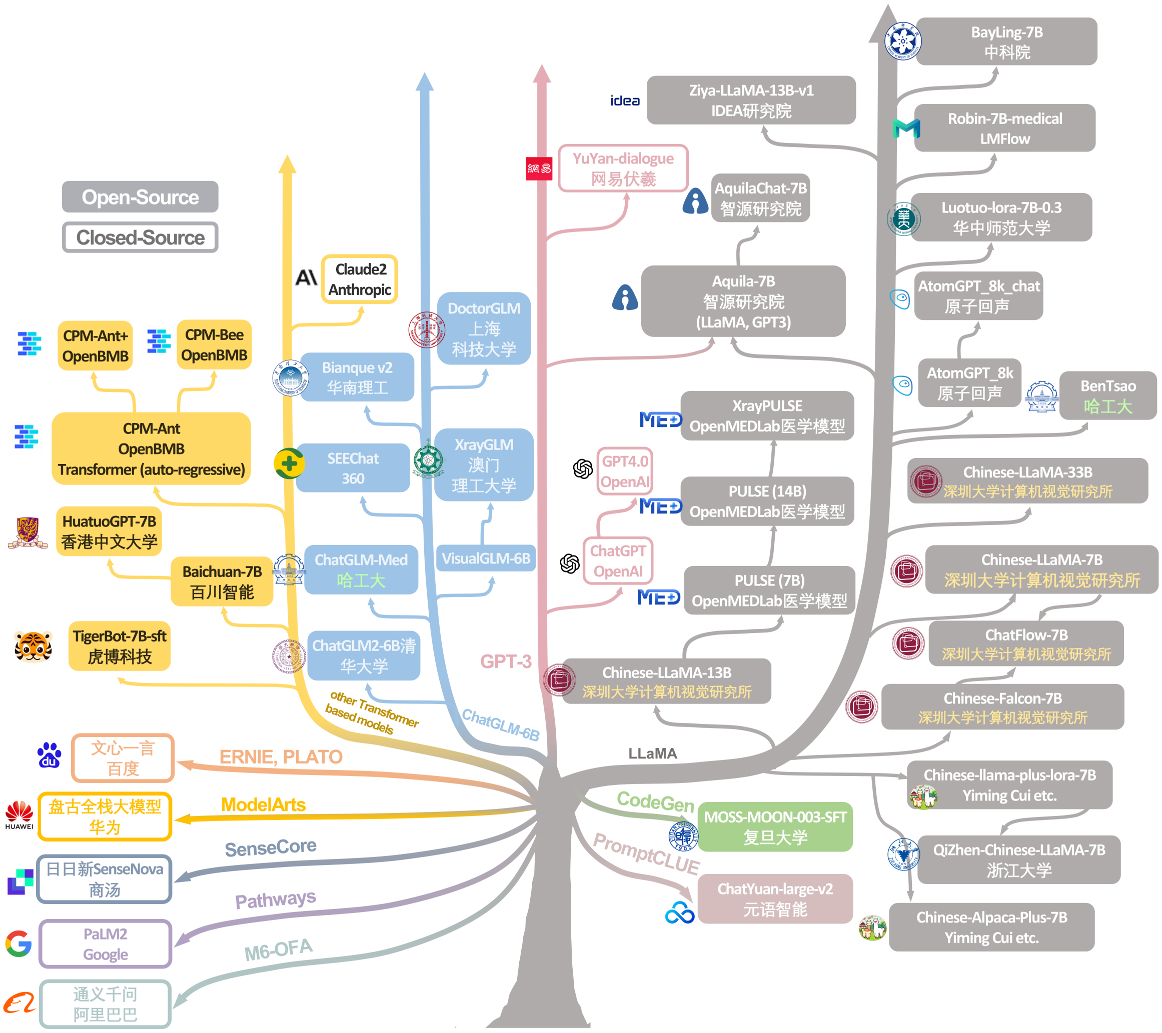


Figure 6.3. Major world-wide LLMs and their relationships.

Table 6.1. Test Results for Compared LLMs

Model	OpenI									MIMIC-CXR								
	zero-shot			one-shot			five-shot			zero-shot			one-shot			five-shot		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Luotuo-lora-7B-0.3	0.147	0.0495	0.1262	0.152	0.0561	0.1356	0.4282	0.3238	0.4168	0.2366	0.1229	0.1899	0.0795	0.0132	0.0761	0.2649	0.1424	0.2491
Ziya-LLaMA-13B-v1	0.1101	0.0316	0.0926	0.1502	0.0285	0.1379	0.2794	0.1694	0.2702	0.2121	0.0968	0.17	0.1103	0.0299	0.0995	0.2806	0.139	0.2512
YuYan-dialogue	0.0622	0.0183	0.0599	0.0702	0.009	0.0627	0.0981	0.0529	0.092	0.0819	0.0303	0.0978	0.0557	0.0123	0.0607	0.1193	0.0807	0.1231
BenTsao	0.0804	0.0368	0.0711	0.0912	0.0201	0.0977	0.1302	0.0803	0.1231	0.1319	0.0618	0.1126	0.0604	0.0423	0.0627	0.1687	0.0807	0.1667
XrayGLM	0.0622	0.0183	0.0599	0.0808	0.0063	0.0631	0.0612	0.0215	0.0501	0.1104	0.0468	0.1211	0.0538	0.0233	0.0531	0.1533	0.0621	0.1611
ChatGLM-Med	0.0796	0.0317	0.0661	0.0702	0.025	0.0815	0.1201	0.0713	0.1488	0.1233	0.0586	0.1344	0.0702	0.0335	0.0641	0.1601	0.0806	0.1744
ChatGPT	0.1203	0.037	0.1052	0.1363	0.0421	0.1205	0.4262	0.2961	0.4113	0.2048	0.0996	0.1702	0.2506	0.1183	0.2052	0.3401	0.1871	0.2921
GPT4	0.1171	0.0343	0.0975	0.1079	0.0328	0.0909	0.1357	0.0617	0.1176	0.1995	0.0858	0.1575	0.2114	0.0875	0.163	0.2029	0.0904	0.1607
ChatGLM2-6B	0.1094	0.0331	0.0909	0.0976	0.0237	0.0886	0.215	0.1346	0.2058	0.2042	0.0964	0.1605	0.0974	0.033	0.0844	0.247	0.1254	0.2198
ChatGLM-6B	0.125	0.0398	0.1087	0.1576	0.0511	0.143	0.2568	0.1451	0.2432	0.2051	0.1007	0.1694	0.1439	0.0464	0.1293	0.2766	0.1409	0.2386
QiZhen-Chinese-LLaMA-7B	0.1122	0.0281	0.0965	0.1033	0.0259	0.0823	0.224	0.1461	0.2123	0.2137	0.0877	0.1743	0.1021	0.041	0.0901	0.258	0.1254	0.2198
MOSS-MOON-003-sft	0.1402	0.0341	0.1241	0.1275	0.0242	0.1143	0.2206	0.1088	0.2057	0.2203	0.0914	0.1757	0.1258	0.0363	0.1062	0.2391	0.0999	0.1957
ChatFlow-7B	0.1048	0.0099	0.0978	0.0942	0.0165	0.0829	0.1844	0.0733	0.1722	0.1145	0.022	0.0933	0.105	0.0169	0.0857	0.1401	0.0409	0.1166
CPM-Bee	0.1193	0.0374	0.1048	0.1548	0.0297	0.1465	0.1582	0.0327	0.1501	0.2022	0.0938	0.1686	0.1125	0.0437	0.0993	0.1354	0.051	0.1184
PULSE-7B	0.1286	0.0413	0.1111	0.0885	0.0319	0.0726	0.1198	0.0576	0.1028	0.2559	0.1246	0.2043	0.1827	0.0796	0.1382	0.1298	0.0503	0.0955
Baichuan-7B	0.003	0.0009	0.0028	0.1328	0.0472	0.1172	0.2485	0.1467	0.2379	0.0057	0.0029	0.0042	0.1746	0.0804	0.1456	0.2301	0.1229	0.2032
Chinese-Falcon-7B	0.0518	0.0168	0.0416	0.0465	0.0154	0.0364	0.0378	0.0176	0.0312	0.1119	0.0557	0.0867	0.094	0.0448	0.0721	0.0491	0.0248	0.0381
AtomGPT_8k	0.0287	0.0013	0.0245	0.0064	0.0001	0.0054	0.0024	0.0001	0.0021	0.0309	0.0014	0.0222	0.0035	0.0001	0.0029	0.0082	0.0001	0.0069
ChatYuan-large-v2	0.0845	0.0223	0.0761	0.1005	0.0331	0.0903	0.2031	0.1078	0.1991	0.1353	0.0632	0.1151	0.1879	0.0793	0.15	0.0108	0.004	0.0102
Bianque v2	0.0227	0.0036	0.0222	0.0271	0.0039	0.0264	0.0304	0.0072	0.0291	0.0294	0.0072	0.0281	0.0295	0.0062	0.0285	0.0225	0.0039	0.0212
AquilaChat-7B	0.0948	0.0279	0.0786	0.1079	0.0259	0.0896	0.2288	0.1454	0.2218	0.1885	0.0862	0.1474	0.1419	0.0408	0.1112	0.2084	0.0895	0.1781
Aquila-7B	0.0373	0.0071	0.0334	0.028	0.0057	0.0247	0.026	0.0087	0.0234	0.0707	0.022	0.0576	0.0411	0.011	0.0339	0.0425	0.0119	0.0353
Chinese-Alpaca-Plus-7B	0.063	0.0056	0.0507	0.0492	0.0035	0.0414	0.0552	0.0064	0.048	0.0898	0.0118	0.0663	0.0795	0.0113	0.0608	0.0807	0.0163	0.0642
TigerBot-7B-sft	0.064	0.0102	0.0527	0.1246	0.0284	0.1083	0.1562	0.0415	0.1401	0.1321	0.0331	0.1005	0.1452	0.043	0.1256	0.205	0.0676	0.1667
XrayPULSE	0.0293	0.0011	0.0239	0.0282	0.0018	0.0233	0.0263	0.0011	0.0225	0.0592	0.0061	0.0398	0.0531	0.005	0.0378	0.0488	0.006	0.0388
DoctorGLM	0.0996	0.0329	0.0861	0.1353	0.0463	0.1205	0.1392	0.0576	0.1328	0.1853	0.0916	0.153	0.0664	0.0146	0.0639	0.2116	0.1055	0.1916
Robin-7B-medical	0.0211	0.0043	0.0173	0.0335	0.0101	0.0285	0.0423	0.016	0.0377	0.0497	0.0139	0.0375	0.0323	0.0044	0.0236	0.0474	0.0132	0.0353
PaLM2	0.1386	0.0477	0.1194	0.1557	0.0551	0.1353	0.1386	0.0492	0.1245	0.2749	0.1442	0.2281	0.2711	0.1446	0.2251	0.2397	0.1216	0.2019
Sensenova	0.0634	0.0106	0.051	0.0682	0.0121	0.055	0.0994	0.0214	0.0876	0.1136	0.0304	0.0809	0.1209	0.036	0.0889	0.1401	0.0395	0.0907
Anthropic Claude2	0.2372	0.1259	0.2193	0.1944	0.0888	0.1713	0.4086	0.2755	0.3904	0.3177	0.153	0.256	0.3222	0.1514	0.2626	0.3116	0.1568	0.2548
BayLing-7B	0.1252	0.0389	0.1044	0.1268	0.0467	0.1181	0.4506	0.3452	0.4436	0.2149	0.107	0.1747	0.0851	0.0229	0.0825	0.2901	0.1722	0.2747

6.1.3 Results

This section presents the evaluation results of various large language models (LLMs) on two extensive datasets, OpenI and MIMIC-CXR. The performance of the models was assessed under three distinct shot settings: zero-shot, one-shot, and five-shot. Model performance was evaluated using three key metrics: Recall@1 (R-1), Recall@2 (R-2), and Recall@L (R-L).

6.1.3.1 OpenI Dataset Results

On the OpenI dataset, Anthropic Claude2 excelled in the zero-shot setting, achieving an R-1 score of 0.2372, an R-2 score of 0.1259, and an R-L score of 0.2193. These results notably surpassed those of other models under the same setting. In the one-shot scenario, the model achieving the highest R-1 score was BayLing-7B with 0.1268, followed closely by Luotuo-lora-7B-0.3 and Ziya-LLaMA-13B-v1 with scores of 0.152 and 0.1502, respectively. However, BayLing-7B was the standout performer in the five-shot setting, registering the highest scores across all metrics with an R-1 score of 0.4506, an R-2 score of 0.3452, and an R-L score of 0.4436.

6.1.3.2 MIMIC-CXR Dataset Results

The evaluation on the MIMIC-CXR dataset showed that the Anthropic Claude2 model retained its superior performance in the zero-shot setting, achieving an R-1 score of 0.3177, an R-2 score of 0.153, and an R-L score of 0.256. PaLM2 emerged as the leading model in the one-shot setting, delivering an R-1 score of 0.2711, an R-2 score of 0.1446, and an R-L score of 0.2251. In the five-shot scenario, the BayLing-7B model continued to outperform other models with the highest R-1 score of 0.2901, R-2 score of 0.1722, and R-L score of 0.2747.

However, some models like AtomGPT_8k registered considerably lower performance across all shot settings and both datasets. For example, AtomGPT_8k scored remarkably low in the OpenI zero-shot setting, with an R-1 score of 0.0287. It continued to score low across other shot settings and in the MIMIC-CXR dataset.

In conclusion, this evaluation underscores the significant diversity in the capabilities of different LLMs, emphasizing the need for careful model selection for specific tasks. The performance variance across different shot conditions has important implications for task-specific LLM selection in future research and applications.

6.1.4 Discussion

6.1.4.1 Impact and Insights

The present study has conducted one of the most exhaustive assessments of world-wide LLMs, focusing primarily on their utilization within the domain of radiology. The meticulous evaluation of these models, using extensive radiology report datasets and juxtaposing them with established global leading models, provides significant insights into their capabilities, limitations, and potential roles within the healthcare sector.

Our findings underscore that multiple LLMs perform comparably in interpreting radiology reports. This alignment points to their advanced natural language understanding skills and highlights their potential utility in enhancing radiology practice, where they can aid in automating radiological image interpretation, assisting in preliminary diagnosis, and thereby freeing up time for healthcare professionals. This is particularly beneficial in regions with limited access to radiologists or in healthcare scenarios where high volumes and time constraints pose significant challenges.

6.1.4.2 Inter-model Differences and Implications

While the performance of the world-wide models showed broad alignment, our results also spotlighted some disparities between the different models. This variance in strengths and weaknesses indicates that the choice of an LLM for a specific application should depend on the particular requirements of that task. Hence, a more profound understanding of these models, to which our study contributes, is critically essential for their effective deployment in the field.

6.1.4.3 Implications of Evaluation Metrics

The evaluation metric adopted in our study is Rouge Score, an N-gram-based method that inherently measures how well models conform to set answers. GPT-4, a universally recognized powerful model, did not outperform its counterpart, ChatGPT, nor did it surpass other models in the Rouge Score. This discrepancy invites a questioning of the significance of Rouge Score as a measure of radiology knowledge.

The BayLing model, for instance, tended to produce succinct answers which, despite their brevity, may be of high quality and accuracy. On the contrary, GPT-4 may be more verbose and consider issues more comprehensively, showing some level of distrust in the input. The difference in results highlights the need to carefully interpret the evaluation scores, taking into account the unique characteristics of each model.

6.1.4.4 Model Size and Performance

Our analysis reveals that to achieve high performance in this specific task, there is no strict need for large models. Models with 7B parameters can produce impressive results, suggesting that we might be on the verge of a fourth industrial revolution driven by these more accessible, lightweight models. This prompts a reconsideration of the belief that model performance is strongly correlated with the size of the model. In fact, smaller models also demonstrated strong capabilities, raising the question of whether intelligence truly arises from the number of parameters and data accumulation.

6.1.4.5 Multimodal LLMs: The Next Frontier

The advent of multimodal LLMs, capable of managing multiple forms of input such as text and images, creates fascinating prospects for future research. Evaluating these models' aptitude to directly interpret radiological images, in addition to textual reports, could revolutionize radiology practice. These multimodal models could find uses in areas like disease detection and diagnosis, treatment planning, and patient monitoring.

6.1.5 Conclusion

In this comprehensive study, we rigorously evaluated the performance of 32 significant world-wide LLMs in the healthcare and radiology sector, comprising both global leading models such as ChatGPT, GPT-4, PaLM2, Claude2 and a robust suite of LLMs developed in other countries such as China. The overarching goal of this exploration was to benchmark these models in the context of interpreting radiology reports, enabling a nuanced understanding of their diverse capabilities, strengths, and weaknesses. Our findings affirm the competitive performance of many Chinese LLMs against their global counterparts, emphasizing their untapped potential in healthcare applications, particularly within radiology. This suggests a trajectory towards a future where these multilingual and diverse LLMs contribute to an enhanced global healthcare delivery system.

Looking ahead, our large-scale study's insights offer a compelling foundation for further exploratory research. There is immense scope for expanding these LLMs into different medical specialties and developing multimodal LLMs, the latter of which could handle complex and diverse data types to provide a more comprehensive understanding of patient health. However, as we navigate this evolving landscape of LLMs, it is imperative to give due consideration to their effective application and ethical deployment. In conclusion, our study hopes to catalyze further exploration and discussion, envisioning an era where LLMs significantly aid in healthcare provision and contribute to an enhanced standard of global patient care.

6.2 Exploring the Trade-Offs: Unified Large Language Models vs Local Fine-Tuned Models for Highly-Specific Radiology NLI Task

Recently, ChatGPT and GPT-4 have emerged and gained immense global attention due to their unparalleled performance in language processing. Despite demonstrating impressive capability in various open-domain tasks, their adequacy in highly specific fields like radiology remains untested. Radiology presents unique linguistic phenomena distinct from open-domain data due to its specificity and complexity. Assessing the performance of large language models (LLMs) in such specific domains is crucial not only for a thorough evaluation of their overall performance but also for providing valuable insights into future model design directions: whether model design should be generic or domain-specific. To this end, in this study, we evaluate the performance of ChatGPT/GPT-4 on a radiology NLI task and compare it to other models fine-tuned specifically on task-related data samples. We also conduct a comprehensive investigation on ChatGPT/GPT-4's reasoning ability by introducing varying levels of inference difficulty. Our results show that 1) GPT-4 outperforms ChatGPT in the radiology NLI task; 2) other specifically fine-tuned models require significant amounts of data samples to achieve comparable performance to ChatGPT/GPT-4. These findings demonstrate that constructing a generic model that is capable of solving various tasks across different domains is feasible.

6.2.1 Datasets

In this section, we present the details of converting the RadQA dataset [228] into the proposed RadQNLI dataset and provide statistical analysis of the resulting RadQNLI dataset.

6.2.1.1 RadQA

The RadQA dataset [228] is a radiology question-answering dataset released in 2022, containing 6,148 question-answer evidence pairs in the radiology domain. It is derived from a random sample of realistic radiology reports from the MIMIC-III database [229] at the patient level and then partitioned into training, validation, and test sets with a ratio of 8:1:1. In addition to the radiology reports, the dataset also includes clinical referrals authored by physicians who ordered the radiology examinations. These referrals contain actual information needs and were assembled alongside the corresponding radiology reports. To generate questions that reflect the information needed in the referrals, two annotators with medical expertise were employed. They annotated answer spans in the radiology reports based on the referrals, resulting in questions that require extensive reasoning and domain-specific knowledge to answer. In general, this dataset is well-suited for conversion into a QNLI dataset for the radiology domain.

6.2.1.2 RadQNLI

Drawing inspiration from previous work on transforming QA datasets into NLI datasets [230, 231, 232], we convert the RadQA dataset into the proposed RadQNLI dataset in a similar way, with the goal to determine whether the context sentence contains the answer to the question. As depicted in the top panel of Fig. **??**, sentences are extracted from the context and then paired with the corresponding question of this context. The labels are automatically assigned for each pair, with a label of 'entailment' assigned if the context sentence contains the answer to the question, and 'not entailment' assigned otherwise. If the sentence contains the answer, we concatenated it with subsequent sentences to ensure the full answer span was included.

To improve the dataset's quality, we introduced a selection process that uses a predefined ROUGE-1 score [233] to screen out sentences with lower similarity (lower lexical overlap) to the questions. Our analysis shows that the ROUGE-1 score between the question and the "entailment" sentences is generally high, indicating a high level of similarity. Conversely, the ROUGE-1 score between the question and the "not entailment" sentences is typically low, indicating a lower level of similarity. A larger ROUGE-1 score will screen out more "not entailment" sentences that have a lower similarity with the question, leading to preserving fewer "not entailment" samples. This can make the NLI task more challenging as both the "entailment" and remaining "not entailment" sentences will have higher similarity to the question, making it harder for the model to distinguish between them. To ensure a suitable difficulty level and balance between the two categories, we selected a ROUGE-1 score of 0.2 as the threshold for lexical overlap between the context sentence and the question. By doing so, we aimed to maintain the challenging nature of the RadQNLI, thereby making it appropriate for training and evaluating advanced models with robust reasoning capabilities.

Table 6.2. Sample Size of RadQNLI Dataset

Measure	Raw	Threshold with R1≥0.2
Avg question tokens	12.34	12.32
Avg sentence tokens	21.66	20.66
# of unique questions	2771	2614
# of unique sentences	11829	5355
# of entailment	4391	4391
# of not entailment	35504	5447
Total samples	39895	9838

Table 6.3. Modalities and Organs Covered by RadQNLI Dataset

Top 5 modalities	Percentage	Top 5 organs	Percentage
X-Ray	12.33%	Lung/chest	29.45%
Computed Tomography (CT)	4.61%	Vessel	11.26%
Magnetic Resonance (MR)	4.07%	Brain	9.09%
CT Angiography (CTA)	3.13%	Abdomen	8.94%
Unknow	77.35%	Cardiac	5.51%

Table 6.2 and Table 6.3 presents the specifics of the newly created RadQNLI dataset, which comprises 2614 questions and 5355 sentences. From these questions and sentences, we generated 9838 question-sentence pairs, with 4391 labeled as "entailment" and 5447 labeled as "not entailment". These samples covers various modalities of imaging information for multiple organs.

6.2.2 Method

In this section, we present the methodology employed for the radiology QNLI task utilizing ChatGPT and GPT-4. We implemented ChatGPT and GPT-4 alongside various prompt designs for natural language inference and contrasted the outcomes with several baseline models.

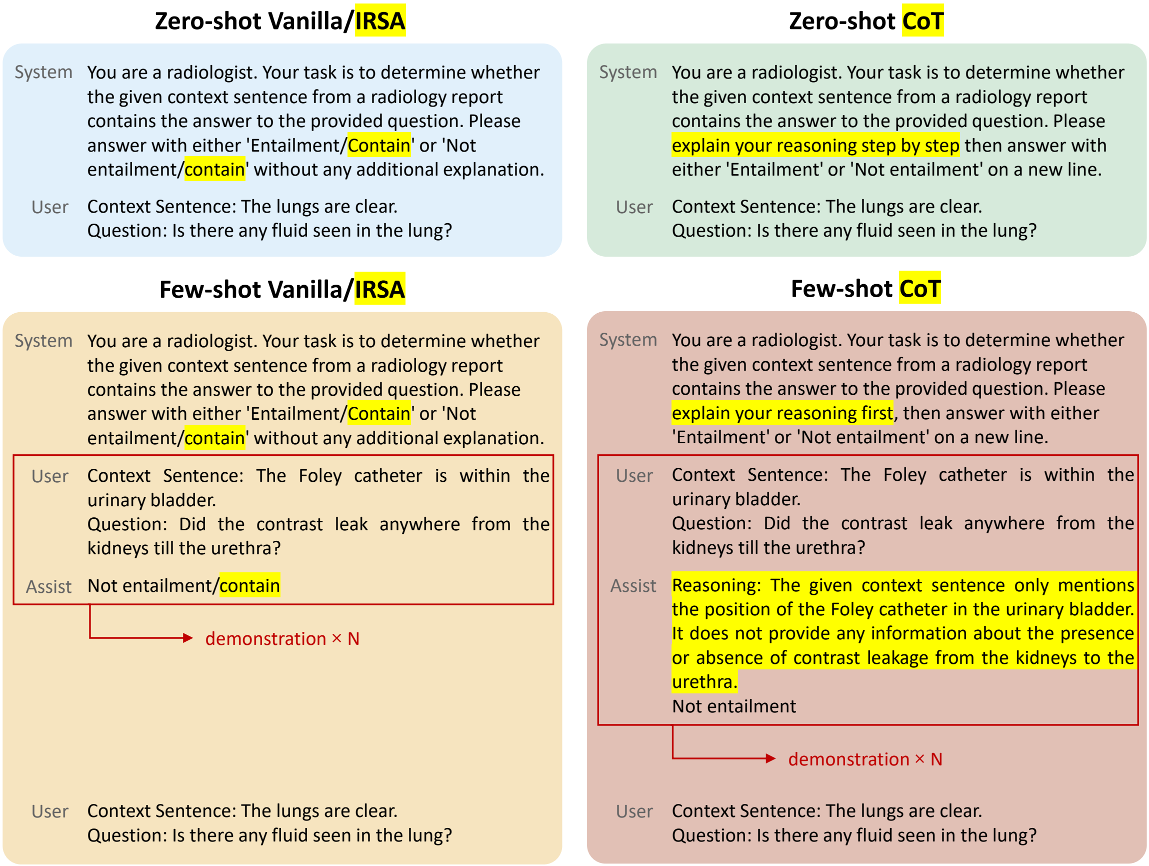
6.2.2.1 Zero-shot and Few-Shot Entailment Inference

The zero-shot and few-shot in-context learning capabilities of ChatGPT and GPT-4 obviate the need for supervised fine-tuning, facilitating rapid adaptation to our novel Radiology QNLI task to evaluate the models' context awareness and understanding in the radiology domain.

Zero-Shot: We formulated the zero-shot prompt using only the task instructions and context sentence-question pairs, which directly requested ChatGPT and

GPT-4 to determine if it was an entailment or not without providing any labeled examples. The prompt for zero-shot inference is presented in Fig.6.4.

Few-Shot: We constructed the few-shot prompt by incorporating 1) task instructions and 2) 10 context sentence-question pairs with corresponding labels as examples for in-context learning. The context-question pair under evaluation was appended at the end. The prompt for few-shot inference is presented in Fig. 6.4.



Zero-shot Vanilla/IRSA

System You are a radiologist. Your task is to determine whether the given context sentence from a radiology report contains the answer to the provided question. Please answer with either 'Entailment/Contain' or 'Not entailment/contain' without any additional explanation.

User Context Sentence: The lungs are clear.
Question: Is there any fluid seen in the lung?

Zero-shot CoT

System You are a radiologist. Your task is to determine whether the given context sentence from a radiology report contains the answer to the provided question. Please explain your reasoning step by step then answer with either 'Entailment' or 'Not entailment' on a new line.

User Context Sentence: The lungs are clear.
Question: Is there any fluid seen in the lung?

Few-shot Vanilla/IRSA

System You are a radiologist. Your task is to determine whether the given context sentence from a radiology report contains the answer to the provided question. Please answer with either 'Entailment/Contain' or 'Not entailment/contain' without any additional explanation.

User Context Sentence: The Foley catheter is within the urinary bladder.
Question: Did the contrast leak anywhere from the kidneys till the urethra?

Assist Not entailment/contain

demonstration × N

User Context Sentence: The lungs are clear.
Question: Is there any fluid seen in the lung?

Few-shot CoT

System You are a radiologist. Your task is to determine whether the given context sentence from a radiology report contains the answer to the provided question. Please explain your reasoning first, then answer with either 'Entailment' or 'Not entailment' on a new line.

User Context Sentence: The Foley catheter is within the urinary bladder.
Question: Did the contrast leak anywhere from the kidneys till the urethra?

Assist Reasoning: The given context sentence only mentions the position of the Foley catheter in the urinary bladder. It does not provide any information about the presence or absence of contrast leakage from the kidneys to the urethra.
Not entailment

demonstration × N

User Context Sentence: The lungs are clear.
Question: Is there any fluid seen in the lung?

Figure 6.4. Prompts used in the experiments. Modifications for IRSA and CoT are highlighted.

6.2.2.2 Inference with Chain of Thought

Prior research has indicated that reasoning tasks can be significantly enhanced by employing Chain of Thought (CoT) prompting, which requests the model to think step by step like a human, resolving multi-step complex reasoning questions sequentially to reduce difficulty and guide the model to follow the chain of thought to ultimately arrive at the answer. We employed the CoT scheme in both zero-shot and few-shot inference.

Zero-Shot with CoT: Inspired by previous work, we included “Please explain your reasoning first and then answer” in the task instruction prompt as presented Fig.6.4.

Few-Shot with CoT: To obtain the CoT for the 10 provided examples, we utilized GPT-4 to auto-generate the reasoning steps, as GPT-4 excels at tasks requiring advanced reasoning and complex instruction comprehension. In cases where GPT-4 output the incorrect label, we prompted the model to regenerate the correct one. The auto-generated reasoning was subsequently refined by a human to form the CoT. The refined CoT for each example, along with the label, was provided as the desired output. The few-shot with CoT prompt is presented in Fig. 6.4.

6.2.2.3 Instruction-Response Semantic Alignment

The original labels of our RadQNLI dataset are ’entailment’ and ’not entailment’, identical to the QNLI dataset. We instructed models to respond with simpler and more common words: either ”contain” or ”not contain”, as opposed to the original label, to better align the semantics in the task instruction prompt and further reduce difficulty. After collecting the responses for each test pair, we replaced them with the original labels back for evaluation.

6.2.2.4 Baseline Models

To compare with ChatGPT and GPT-4, we tested the following baseline models in two categories.

Text Generation Models. For these text generation models, which possess the ability of in-context learning, we employed the same prompt as ChatGPT and GPT-4 to request models to perform inference on the test set.

1. **LlaMA:** LLaMA [234] (Large Language Model Meta AI) is a series of foundational large language models released by Meta AI in February 2023. Researchers trained these models, ranging from 7 billion to 65 billion parameters (7B, 13B, 33B, and 65B), on trillions of tokens. The 65B model maintains a relatively smaller size while being competitive with GPT-3.5 and other top LLMs.
2. **Alpaca:** Alpaca [235] is an instruction-following model released by Stanford University. It was fine-tuned from Meta's LLaMA 7B model and trained on 52K instruction-following demonstrations generated using text-davinci-003 dataset.Despite being a small and easily reproducible model, Alpaca performs comparably to text-davinci-003 in terms of performance.
3. **LlaMA 2:** LLaMA 2 [236] is built upon its predecessor LLaMA, with 40% more training data, and has double the context length than Llama 1. The fine-tuned chat version LLaMA 2 has been trained on over 1 million human annotations for better human preference alignment. Evaluation on several external benchmarks including reasoning, coding, proficiency, and knowledge tests show that LLaMA 2 outperforms not only LLaMA 1 but also other LLMs.
4. **Radiology LlaMA 2:** Radiology LLaMA 2 [237] is an instruction-following model fine-tuned from LLaMA 2. Utilizing the MIMIC-CXR dataset for instruction tuning, this model leverages specialized knowledge in radiology to en-

hance performance, thereby excelling in summarizing Findings into Impressions in radiology reports.

5. **Bloomz:** BLOOMZ [238] is a family of models capable of following human instructions in dozens of languages zero-shot. They were generated by fine-tuning BLOOM [239] pretrained multilingual language models on crosslingual task mixture dataset(xP3) [238]. Bloomz has been found the ability of crosslingual generalization and performing effectively on tasks or languages that were previously unseen.

BERT-Based Fine-Tuned Models. We utilized pretrained models, fine-tuned them on the training and validation sets, and then evaluated their performance on the test set for comparison.

1. **BERT:** In 2018, Google released Bidirectional Encoder Representations from Transformers (BERT) [179], a collection of masked-language models. BERT stands out as the first unsupervised language representation model that is deeply bidirectional, meaning it is capable of analyzing language in both directions. This capability enables BERT to extract high-quality language features that can be fine-tuned on domain-specific data for improved performance. Notably, BERT's pre-training is conducted solely on plain text corpora.
2. **DistilBERT:** DistilBERT is a compact and efficient Transformer model based on BERT. By distilling BERT, DistilBert possesses a smaller parameter count than bert-base-uncased by 40% and performs faster by 60%. DistilBERT achieves comparable performance to BERT on the General Language Understanding Evaluation (GLUE) language understanding benchmark, maintaining an accuracy rate of over 95%.
3. **RoBERTa:** The RoBERTa model [240] was based on BERT and share same architecture. By adding dynamic masking, removing the next-sentence pre-

training objective and training with larger batches, RoBERTa is a better reimplementation of BERT with some modifications to the key hyperparameters.

4. **BioBERT:** BioBERT (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) [241] is a domain-specific language representation model pre-trained on large-scale biomedical corpora PubMed abstracts (PubMed) and PubMed Central full-text articles (PMC). When compared to BERT and other pre-existing models, BioBERT demonstrates superior performance in various biomedical text mining tasks.
5. **ClinicalBERT:** ClinicalBERT [242], which is based on BERT, utilizes bidirectional transformers to analyze clinical notes from the Medical Information Mart for Intensive Care III (MIMIC-III) dataset. The model has been evaluated by medical professionals and has been found to establish high-quality relationships between medical concepts, as evidenced by their judgements.

6.2.2.5 Implementation

The ChatGPT model used in this paper is 'gpt-3.5-turbo-0301', released on March 1st, 2023. The GPT-4 model employed in this paper is 'gpt-4-0314', released on March 14th, 2023. The baseline pretrained models and corresponding experiments were conducted using the Hugging Face Transformers package on an NVIDIA A100 GPU.

6.2.2.6 Evaluation Metrics

We employed accuracy as the evaluation metric to assess the models' reasoning and inference performance. All experiments involving ChatGPT and GPT-4 models report the average accuracy with the standard deviation of three repetitions to eliminate randomness and improve reproducibility.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

6.2.3 Experiment and Analysis

6.2.3.1 ChatGPT and GPT-4 with Different Prompting

We assess the inference performance of ChatGPT and GPT-4 using three techniques: zero-shot and few-shot in context learning, chain of thought prompting, and instruction-response semantic alignment prompting. Fig. 6.5 displays the inference accuracy of ChatGPT and GPT-4 under various settings, with zero-shot results on the left and few-shot results on the right. In each setting, we report the outcomes for four types of prompting designs: the blue bars signify the vanilla prompt design, the green bars represent the vanilla prompt with chain of thought, the yellow bars indicate the vanilla prompt with instruction-response semantic alignment, and the red bars denote the vanilla prompt with both CoT and IRSA. Detailed prompts for the different prompting techniques can be found in Fig. 6.5. As depicted in Fig. 6.5, GPT-4 outperforms ChatGPT in all four prompting designs for both zero-shot and few-shot settings, showcasing a powerful capability for human-level understanding in language tasks [243]. Additionally, we provide an ablation analysis of the three techniques below.

Zero-shot and Few-shot Prompting: Based on the results using the vanilla prompt design (blue bars), the few-shot setting considerably surpasses zero-shot prompting for both ChatGPT and GPT-4, with accuracy increasing from 0.6390 to 0.7380 and from 0.6813 to 0.8315, respectively. The improvement of over 0.10 demonstrates that few-shot prompting, by simply providing demonstrations during inference, is an efficient and effective technique for directly enhancing model performance on the QNLI task. Conversely, GPT-4 improves accuracy by 0.423 in the

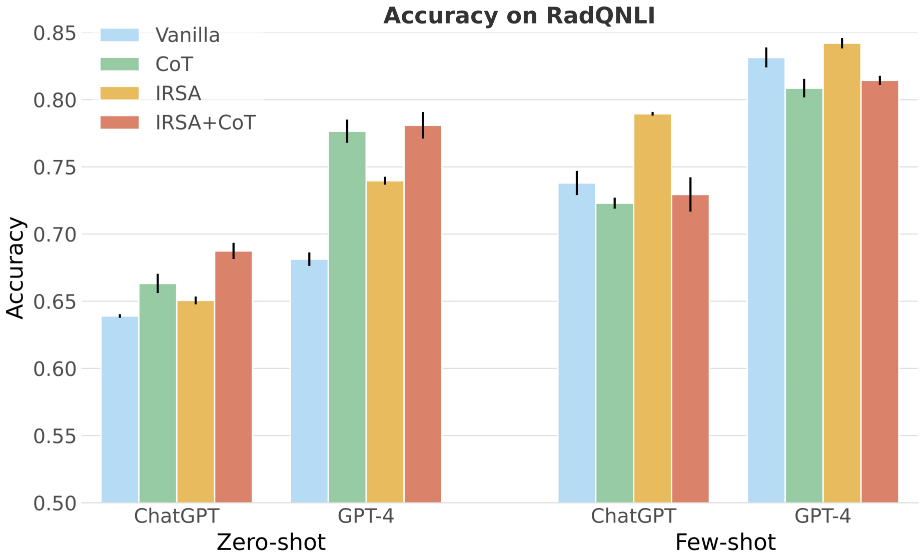


Figure 6.5. We assessed the impact of various prompt methods (as detailed in Sec. 6.2.3.1) on the performance of ChatGPT and GPT-4 for both zero-shot and few-shot scenarios..

zero-shot setting, while in the few-shot setting, the improvement reaches 0.935, nearly doubling compared to the zero-shot setting. This significant enhancement in the few-shot setting indicates that GPT-4 can better comprehend and utilize the examples in the prompt, serving as a guide to retrieve and revise pre-trained knowledge and generate desired responses that are highly aligned and consistent with the examples.

Chain of Thought: The results of experiments utilizing chain of thought prompting are represented by the green bars in Fig. 6.5. In the zero-shot setting, as displayed on the left side of Fig. 6.5, compared to the vanilla prompt design, CoT prompting enhances accuracy for both ChatGPT and GPT-4 by 0.243 and 0.952, respectively. The differing percentages confirm that GPT-4 is better facilitated by CoT in the zero-shot setting, as its exceptional reasoning ability can be effectively coupled with CoT. However, in the few-shot setting, shown on the right side of Fig. 6.5, there is a roughly 0.2 decrease in prediction accuracy for both ChatGPT and GPT-4. A possible explanation for the contrasting effects of CoT on zero-shot and few-shot settings is that in the zero-shot setting, without examples to follow, the model must rely solely on pre-trained knowledge to generate responses. In this scenario, CoT enables the model to break down complex tasks into a chain of smaller, more manageable tasks and solve them sequentially. This approach allows the model to accurately retrieve specific pre-trained knowledge for each smaller task and better utilize reasoning and context awareness abilities, leading to a significant improvement in accuracy. Conversely, in the few-shot setting, although the provided examples can help the model better understand complex tasks by demonstrating desired reasoning steps and corresponding responses, the model may also overfit on the examples shown in the prompt, particularly when the model is pre-set to focus on deterministic, realistic data with less creativity and randomness. Furthermore, CoT prompting in the few-shot setting significantly increases the number of tokens in the prompt, which in turn increases the complexity of the task. The model may concentrate on the examples and neglect the essential instruction obscured by the numerous examples.

Instruction-Response semantic alignment: The results of experiments using Instruction-Response Semantic Alignment (IRSA) prompting are indicated by the yellow bars in Fig. 6.5. Compared to the vanilla prompt design, IRSA prompting outperforms in all four scenarios—zero-shot or few-shot settings, with or without IRSA—increasing prediction accuracy by an average of 0.330. When compared to CoT prompting, in the zero-shot setting, IRSA prompting achieves competitive performance. In the few-shot setting, while CoT prompting falls short, using IRSA prompting alone enables both ChatGPT and GPT-4 models to achieve their best performance at 0.7895 and 0.8420, respectively. This suggests that IRSA prompting is an effective and efficient technique for QNLI tasks, as it eliminates the additional cost of unnecessary reasoning components in responses when using paid LLMs like ChatGPT and GPT-4.

6.2.3.2 Comparison with recently released LLMs

We report the results of different large-scale language models on the RadQNLI dataset in Table 6.4. The findings demonstrate that ChatGPT and GPT-4 explicitly outperform all other recently released LLMs. Specifically, LLaMA exhibits the lowest accuracy, often repeating the instruction or generating random text, thus failing to align with human preferences. In contrast, other LLMs output the predicted label in expected format. However, Bloomz and LLaMA 2 consistently respond with "Entailment" in all tests, making the accuracy results meaningless in this binary classification task. Meanwhile, Alpace and Radiology LLaMA 2 produce well-structured responses, though their accuracy remains significantly lower than that of ChatGPT and GPT-4. This difference is attributed to Alpaca and Radiology LLaMA 2 being instruction-tuned on LLaMA, which better aligns them with human response preferences. Furthermore, Radiology LLaMA 2's incorporation of domain-specific knowledge from the MIMIC-CXR dataset enhances its performance over Alpaca.

In summary, ChatGPT and GPT-4, as opposed to open-source LLMs, demonstrate enhanced language understanding capabilities in the medical domain and a significantly better alignment with human preferences.

6.2.3.3 Comparison with BERT-based Fine-tuning

The proposed RadQNLI dataset consists of 7,911 training samples. We randomly select 10, 100, 500, 1,000, and 7,911 training samples from the original training set and conduct five fine-tuning experiments with different training set sizes to explore the effect of training set size on the performance of fine-tuned BERT-based models in QNLI tasks and the feasibility of fine-tuning BERT-based models in few-shot-like settings. The entire validation and test sets are used in these experiments. The results,

Table 6.4. The accuracy(ACC) of the five recent LLMs tested on proposed RadQNLI dataset with zero-shot incontext learning. Red and blue denotes the best and the second-best results, respectively.

Methods	ACC
Bloomz (7b)	0.5109
Alpaca	0.5435
LLaMA (7b)	0.2107
LLaMA 2 (7b)	0.4708
Radiology LLaMA 2 (7b)	0.5558
ChatGPT	0.6875
GPT-4	0.7809

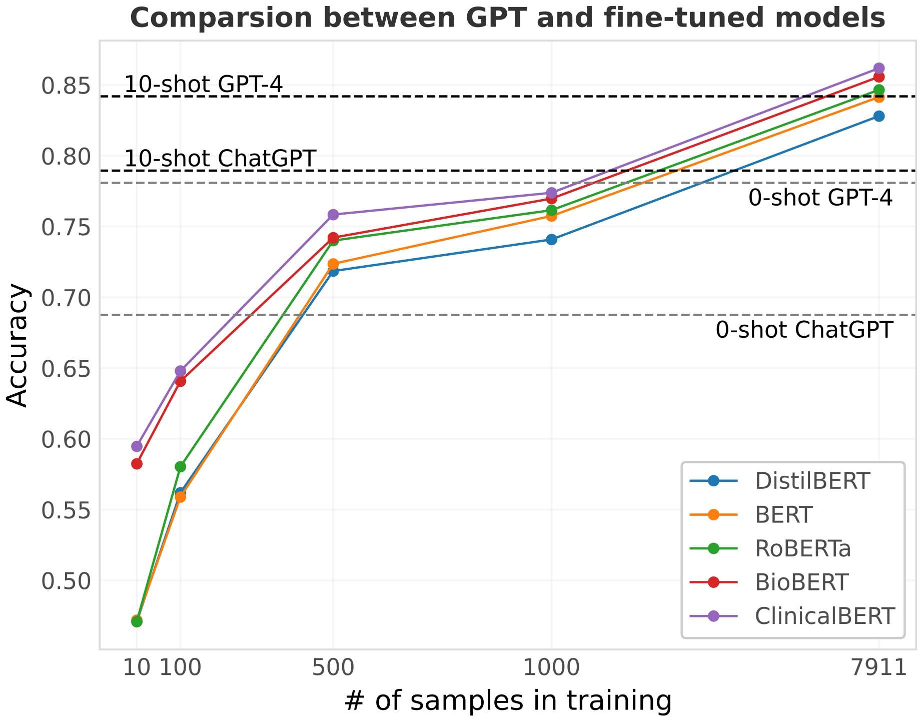


Figure 6.6. We compared the performance of ChatGPT, GPT-4, and five other fine-tuned LLMs. To evaluate the impact of prompt examples on ChatGPT and GPT-4, we tested their performance with zero-shot and 10-shot examples, where greater example usage resulted in higher accuracy. Additionally, we explored the effect of increasing the size of the training dataset for the fine-tuned LLMs, which led to improved accuracy..

along with the best performances of ChatGPT and GPT-4, are displayed in Fig. 6.6 for comparison. With the increment of training set size, the prediction accuracy of all BERT-based models continuously improves, from an average accuracy of 0.5181 across five models with 10 training samples to 0.8467 with the full training set. Using the exact same 10 training samples as in the fine-tuning experiment as examples in few-shot prompting, ChatGPT achieves a competitive accuracy of 0.7895. GPT-4 attains an accuracy of 0.8420, very close to ClinicalBERT, the best BERT-based model fully fine-tuned on the RadQNLI dataset. These results suggest that ChatGPT and GPT-4 are more efficient than traditional BERT-based fine-tuning by utilizing significantly fewer labeled data and eliminating the fine-tuning process altogether.

6.2.3.4 Difficulty Level of RadQNLI

To further assess the difficulty level of the RadQNLI dataset, we experimented with different parameter settings of ROUGE-1 for thresholding lexical overlap, as described in the Dataset section. A higher threshold of lexical overlap between the context sentence and the question can increase the inference difficulty on the dataset because a higher ROUGE-1 score implies that the context sentence and the question either have a similar structure or describe the same thing, making it challenging to determine whether the sentence entails the answer to the question or is merely related to the question. In contrast, a low ROUGE-1 score indicates that the context sentence and the question are not related at all, and therefore, not entailment. It is worth noting that sentence-question pairs with labeled entailment have inevitably higher ROUGE scores; therefore, all samples with entailment labels are pre-included, and the threshold is only applied to non-entailment samples. We evaluate the ChatGPT model with a 3-shot setting on the test set generated with different lexical overlap thresholds. The results in Table 6.5 show that inference accuracy increases with lower lexical overlap thresholds, indicating that by modifying the threshold, we can easily control the difficulty level of the generated dataset, which can be useful for assessing different parameter scale LLMs on a reasonably challenging dataset.

Table 6.5. We adopted ROUGE scores to set different levels of inference difficulty and assessed ChatGPT's performance on tasks of varying levels of difficulty.

Sample	Class distribution in test set		ACC
	entail	not entail	
R1≥0.3	460	193	0.6186±0.0054
R1≥0.2	460	517	0.6986±0.0066
R1≥0.1	460	1327	0.7934±0.0044
R1>0	460	1681	0.8100±0.0023

6.2.4 Discussion and Conclusion

6.2.4.1 Domain-Specific NLI Datasets

Large-scale datasets constructed by crowdworkers have greatly contributed to the progress of domain-agnostic NLI. However, such datasets are prone to statistical artifacts that arise during the annotation process, as previous studies have shown [244, 245, 246]. These artifacts pose a significant challenge in the development and evaluation of NLI models, as they can lead to hypothesis-only classifiers achieving better-than-random performance. To address this challenge, some researchers have attempted to create more robust and challenging datasets [247]. In this study, we propose a new dataset, RadQNLI, specifically designed for the radiology domain, with the help of multiple experts. However, developing a highly specialized dataset requires professional expertise, which makes it difficult to generate large-scale data in a short period. Therefore, our dataset has a smaller volume of data compared

Table 6.6. Monetary costs ($) and inference times (s/item) of ChatGPT and GPT4 on RadQNLI testset.

Method	Length (tokens)		ChatGPT		GPT4	
	prompt	completion	cost	time	cost	time
Zero-shot+IRSA	130.86	2.63	0.26	0.41	3.99	0.90
Zero-shot+CoT	144 .86	66.13	0.41	2.40	8.12	12.45
Few-shot+IRSA	605.86	1.54	1.19	0.43	17.85	0.96
Few-shot+CoT	1007.86	41.77	2.05	1.62	31.99	5.41

to open-domain datasets. To ensure a comprehensive evaluation of the dataset, we conducted a detailed statistical analysis and used ROUGE score to set up tasks with varying levels of inference difficulty during the experiments, in order to filter out possible statistical artifacts as much as possible. In future research, it will remain a crucial issue to incorporate and leverage the knowledge of experts to guide the automation of specialized dataset generation.

6.2.4.2 Uniform vs Domain-Specific Models

Creating a unified, general-purpose model to solve problems in all fields versus creating local specific models for different fields is an important issue. Especially at this particular point in time, where LLMs represented by ChatGPT and GPT-4 have achieved unprecedented success in different tasks, bringing new hope for creating a unified model. In this study, we aimed to explore this issue further by comparing the performance of ChatGPT/GPT-4 and locally fine-tuned models in a highly specialized task - radiology language inference. The results indicated that local models' performance falls short of that of ChatGPT/GPT-4 when the amount of specialized data is limited. However, with sufficient data, local models outperform the unified large models. Therefore, we are faced with the dilemma of choosing between building a unified, general-purpose model or creating effective datasets, both of which require substantial resources when reaching a certain scale (details can be seen in Table 6.6). Alternatively, a more promising approach to this issue is to combine existing unified LLMs with limited specific datasets to better solve tasks in specialized domains.

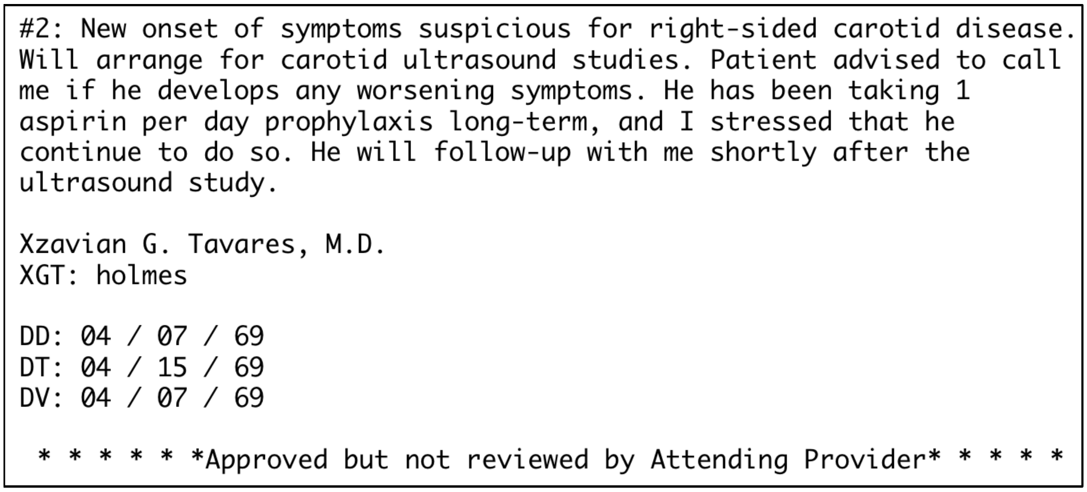
In conclusion, in this work, we evaluated the performance of ChatGPT/GPT-4 on a radiology NLI task and compared it to other models fine-tuned specifically on task-related data samples. The results showed that GPT-4 outperformed ChatGPT in the radiology NLI task, and other specifically fine-tuned models require significant amounts of data samples to achieve comparable performance to ChatGPT/GPT-4. These findings demonstrate that constructing a unified model capable of solving various tasks across different domains is feasible. However, further research is needed to determine whether this approach is preferable to creating domain-specific models, and how to effectively utilize limited specialized datasets with large LLMs.

6.3 DeID-GPT: Zero-shot Medical Text De-Identification by GPT-4

The digitization of healthcare has facilitated the sharing and re-using of medical data but has also raised concerns about confidentiality and privacy. HIPAA (Health Insurance Portability and Accountability Act) mandates removing re-identifying information before the dissemination of medical records. Thus, effective and efficient solutions for de-identifying medical data, especially those in free-text forms, are highly needed. While various computer-assisted de-identification methods, including both rule-based and learning-based, have been developed and used in prior practice, such solutions still lack generalizability or need to be fine-tuned according to different scenarios, significantly imposing restrictions in wider use. The advancement of large language models (LLM), such as ChatGPT and GPT-4, have shown great potential in processing text data in the medical domain with zero-shot in-context learning, especially in the task of privacy protection, as these models can identify confidential information by their powerful named entity recognition (NER) capability. In this work, we developed a novel GPT4-enabled de-identification framework ("DeID-GPT") to automatically identify and remove the identifying information. Compared to existing commonly used medical text data de-identification methods, our developed DeID-GPT showed the highest accuracy and remarkable reliability in masking private information from the unstructured medical text while preserving the original structure and meaning of the text. This study is one of the earliest to utilize ChatGPT and GPT-4 for medical text data processing and de-identification, which provides insights for further research and solution development on the use of LLMs such as ChatGPT/GPT-4 in healthcare.

6.3.1 Datasets

The i2b2/UTHealth Challenge: We benchmark our proposed method using the 2014 i2b2/UTHealth de-identification challenge dataset [248]. Upon request, the Blavatnik Institute of Biomedical Informatics at Harvard University granted us access to this dataset. This dataset contains 1,304 free-form clinical notes of 296 diabetic patients. All PHI entities were manually annotated and replaced with surrogates. Specifically, names, professions, locations, ages, dates, contacts and IDs were replaced by surrogate information to protect privacy and facilitate de-identification research. For example, if there is a real patient named "Mr. James McCarthy" who visited the hospital on 12/01/2013, these strings will be replaced by "Mr. Joshua Howard" and "04/01/2060", respectively. Figure 6.7 presents a sample clip of this data, and figure 6.8 presents the same excerpt de-identified by ChatGPT/GPT-4.



#2: New onset of symptoms suspicious for right-sided carotid disease.
Will arrange for carotid ultrasound studies. Patient advised to call
me if he develops any worsening symptoms. He has been taking 1
aspirin per day prophylaxis long-term, and I stressed that he
continue to do so. He will follow-up with me shortly after the
ultrasound study.

Xzavian G. Tavares, M.D.
XGT: holmes

DD: 04 / 07 / 69
DT: 04 / 15 / 69
DV: 04 / 07 / 69

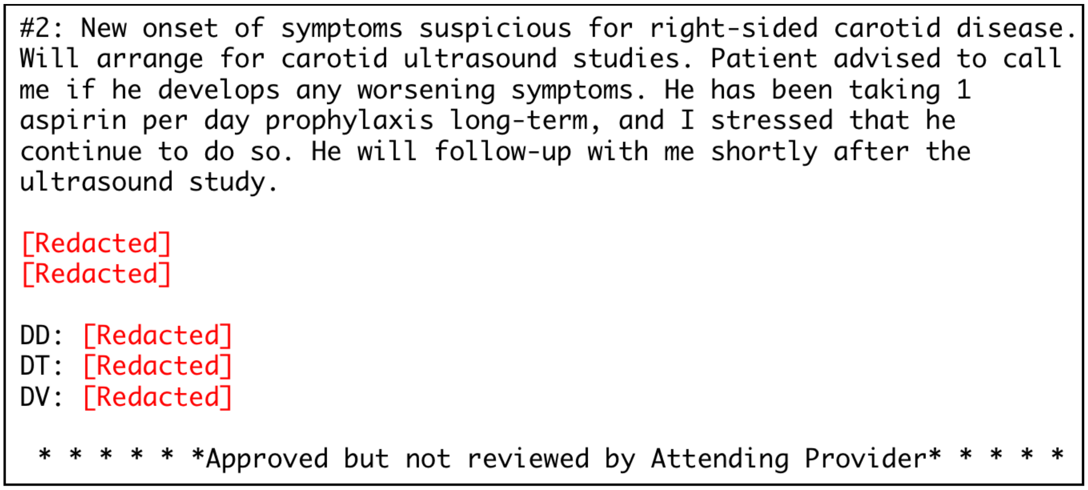
* * * * * *Approved but not reviewed by Attending Provider* * * * *

Figure 6.7. Sample original clinical note from the 2014 i2b2/UTHealth dataset..

6.3.2 Methods

In this section, we will describe the methodology of this work. Our primary approach is to utilize API access and manual testing to evaluate ChatGPT (powered by GPT-3.5) and GPT-4's (through OpenAI's web interface that is shared with ChatGPT)performance on anonymizing clinical notes. We will describe our entire workflow in detail, from the data preprocessing to evaluation.

It is noteworthy to point out that the grand paradigm shift from fine-tuning to prompt-based in-context learning has revolutionized the NLP field. In this study, we intend to devise a new strategy to employ the zero-shot capability of very recent language models to complete the data de-identification process. We carefully design prompts that work well with ChatGPT and GPT-4 to generate the best results with minimal human annotation efforts. Thanks to the scale of LLMs and the power of in-context learning, the presented framework requires no change when being applied to different data. We present a full pipeline that is straightforward to implement and naturally explainable.



#2: New onset of symptoms suspicious for right-sided carotid disease.
Will arrange for carotid ultrasound studies. Patient advised to call
me if he develops any worsening symptoms. He has been taking 1
aspirin per day prophylaxis long-term, and I stressed that he
continue to do so. He will follow-up with me shortly after the
ultrasound study.

[Redacted]
[Redacted]

DD: [Redacted]
DT: [Redacted]
DV: [Redacted]

* * * * * *Approved but not reviewed by Attending Provider* * * * *

Figure 6.8. Sample de-identified clinical note from the 2014 i2b2/UTHealth dataset..

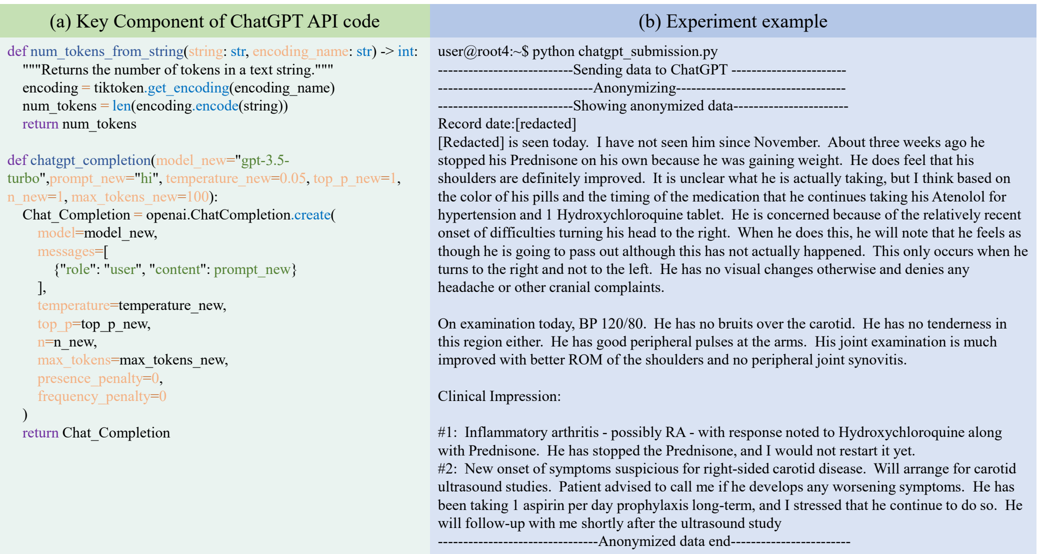
6.3.2.1 Data Preprocessing

The original Harvard 2014 i2b2/UTHealth de-identification challenge dataset is stored as XML files. One XML file corresponds to one complete clinical note that documents the symptoms, clinical records and medical impressions of one particular visit. Such files consist of various XML tags that correspond to different information in the clinical notes.

We have implemented in-house scripts to extract information from these XML files and store them in a reference database. The main text of the clinical notes are further cleaned and stored as text files. These files are the input to the LLMs audited in this study. In addition, the sensitive text entities identified by human annotators are extracted and reserved for performance evaluation.

6.3.2.2 Accessing ChatGPT and GPT-4

To access the OpenAI API, people will need to create an account on OpenAI's websitrate and obtain an API key. Once you have your API key, you can use it to make API requests to OpenAI, including requests to the ChatGPT model. For more detail, please refer to our open-source code at GitHub. Fig. 6.9 displays the crucial code elements of our ChatGPT API. After setting the appropriate parameters and submitting the prompts to the ChatGPT server, the generated texts can be obtained, as indicated in the right column of Fig. 6.9. The left column of Fig. 6.10 depicts the step-by-step procedure for anonymizing sensitive information using our ChatGPT API. The sensitive information contained in the response text (shown in the right column of Fig. 6.10) is already concealed by ChatGPT. Currently, there is no open access to the GPT-4 API. Therefore, we rely on manually testing on the OpenAI web interface to carry out our GPT-4 experiments.


(a) Key Component of ChatGPT API code

```
def num_tokens_from_string(string: str, encoding_name: str) -> int:
    """Returns the number of tokens in a text string."""
    encoding = tiktoken.get_encoding(encoding_name)
    num_tokens = len(encoding.encode(string))
    return num_tokens

def chatgpt_completion(model_new="gpt-3.5-
turbo",prompt_new="hi", temperature_new=0.05, top_p_new=1,
n_new=1, max_tokens_new=100):
    Chat_Completion = openai.ChatCompletion.create(
        model=model_new,
        messages=[
            {"role": "user", "content": prompt_new}
        ],
        temperature=temperature_new,
        top_p=top_p_new,
        n=n_new,
        max_tokens=max_tokens_new,
        presence_penalty=0,
        frequency_penalty=0
    )
    return Chat_Completion
```

(b) Experiment example

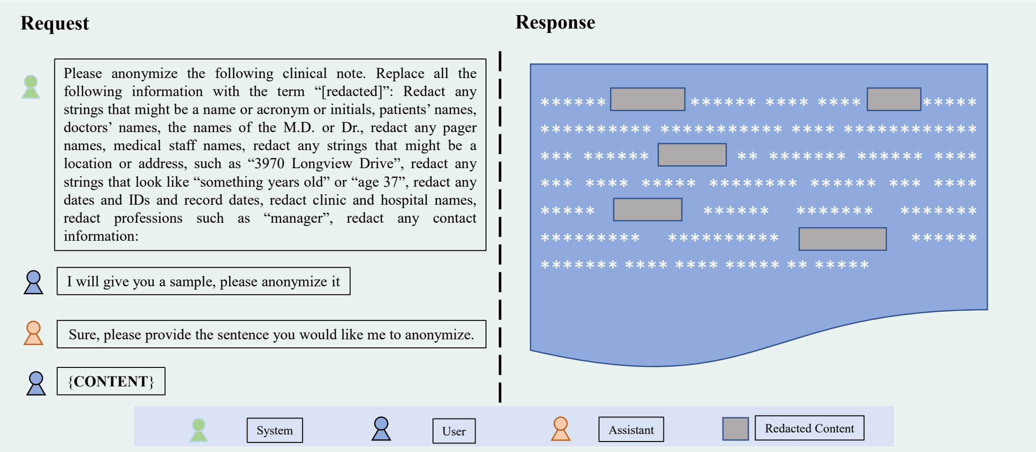
```
user@root4:~$ python chatgpt_submission.py
--------------------------Sending data to ChatGPT ----------------------
------------------------------Anonymizing---------------------------------
--------------------------Showing anonymized data----------------------
Record date:[redacted]
[Redacted] is seen today.  I have not seen him since November.  About three weeks ago he
stopped his Prednisone on his own because he was gaining weight.  He does feel that his
shoulders are definitely improved.  It is unclear what he is actually taking, but I think based on
the color of his pills and the timing of the medication that he continues taking his Atenolol for
hypertension and 1 Hydroxychloroquine tablet.  He is concerned because of the relatively recent
onset of difficulties turning his head to the right.  When he does this, he will note that he feels as
though he is going to pass out although this has not actually happened.  This only occurs when he
turns to the right and not to the left.  He has no visual changes otherwise and denies any
headache or other cranial complaints.

On examination today, BP 120/80.  He has no bruits over the carotid.  He has no tenderness in
this region either.  He has good peripheral pulses at the arms.  His joint examination is much
improved with better ROM of the shoulders and no peripheral joint synovitis.

Clinical Impression:

#1:  Inflammatory arthritis - possibly RA - with response noted to Hydroxychloroquine along
with Prednisone.  He has stopped the Prednisone, and I would not restart it yet.
#2:  New onset of symptoms suspicious for right-sided carotid disease.  Will arrange for carotid
ultrasound studies.  Patient advised to call me if he develops any worsening symptoms.  He has
been taking 1 aspirin per day prophylaxis long-term, and I stressed that he continue to do so.  He
will follow-up with me shortly after the ultrasound study
-------------------------------Anonymized data end-----------------------
```

Figure 6.9. The left column shows the main components of the ChatGPT API code, while the right column shows the steps involved in generating an anonymized example..



Please anonymize the following clinical note. Replace all the following information with the term "[redacted]": Redact any strings that might be a name or acronym or initials, patients' names, doctors' names, the names of the M.D. or Dr., redact any pager names, medical staff names, redact any strings that might be a location or address, such as "3970 Longview Drive", redact any strings that look like "something years old" or "age 37", redact any dates and IDs and record dates, redact clinic and hospital names, redact professions such as "manager", redact any contact information:

I will give you a sample, please anonymize it

Sure, please provide the sentence you would like me to anonymize.

{CONTENT}

Figure 6.10. To anonymize clinical notes using the ChatGPT API, we first supply prompts from the system and then send the clinical notes from the user role. The returned results from ChatGPT will be anonymized by the model, and detected sensitive PHI information will be replaced by the term 'redacted'..

HIPAA Identifiers			
1	Names	10	Account numbers
2	All geographical address elements smaller than state	11	Certificate numbers
3	All data elements related the individual (except year)	12	Vehicle serial numbers and identifiers
4	Phone numbers	13	Device serial numbers and identifiers
5	Fax numbers	14	Web resource locators (URLs) and links
6	Email addresses	15	IP addresses
7	Social security numbers	16	Biometric identifiers (e.g. fingerprint)
8	Medical record numbers	17	Full face photographic images
9	Health plan beneficiary numbers	18	Any unique identifying number, code, or characteristic

Table 6.7. List of HIPAA identifiers.

6.3.2.3 A Simple and Versatile Framework

Our goal is to develop a workflow that greatly simplifies clinical data de-identification, which can facilitate any subsequent research and collaborations. The advent of LLMs such as ChatGPT and GPT-4 has revolutionized NLP, and we see great potential in applying LLMs to the realm of privacy protection. The key advantages of LLMs can be categorized as **Simplicity**, **Annotation-free**, and **Adaptability**. These positive characteristics make LLMs the ideal tools for data anonymization applications.

Specifically, in-context learning eliminates the need for fine-tuning, thereby saving significant amounts of time and reducing pipeline complexity [249, 250]. Supervised fine-tuning is not a trivial task and requires sufficient experience in NLP.

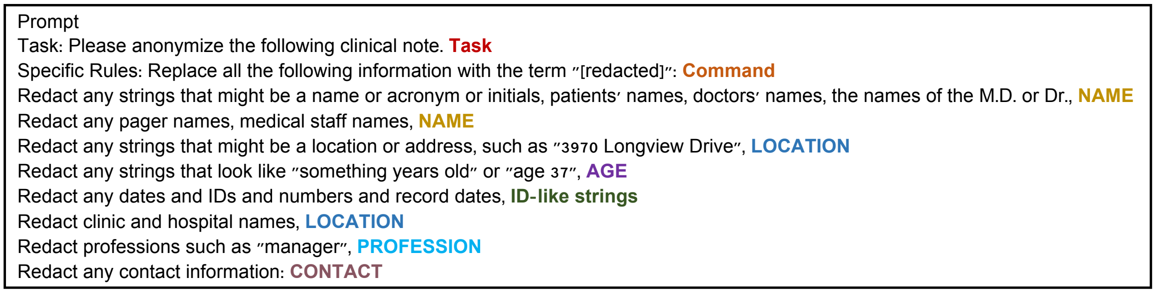
In addition, this approach has exceptional zero-shot and few-shot learning capabilities [189, 250, 251], which can reduce the need for large-scale annotation efforts, as the model can learn from a relatively small amount of annotated data.

By incorporating contextual information, which allows the model to better understand the meaning and context of the data it is processing, in-context learning enables the adaptation to new tasks and domains with great ease [189, 250]. Indeed, LLMs are very versatile and flexible. **Unlike any previous methods, no code or procedural changes are required when applying our solution to different hospitals, different languages and different data formats**.

These advantages make in-context learning a promising approach for a wide range of NLP tasks, including de-identification. We develop this study to methodologically validate ChatGPT and GPT-4's advantages.

Algorithm 1 Framework of DeID-GPT for few-shot text classification.

```
Input: original dataset $D_o$ and HIPAA-compliant prompt $P_h$
Initialize: Initialized ChatGPT $model$
Definition: $D_o$ is original clinical notes dataset, $D_d$ is the de-identified dataset and $D$eId-GPT is the de-identification method based on ChatGPT. HIPAA-compliant prompt $P_h$ is the prompt to guide ChatGPT
Parameters: In-context learning epochs of the original dataset $epoch_o$
  for epoch in $epoch_o$ do
    train($model$, $D_o$, $P_h$)
  end for
  $D_d$ = $DeId\text{-}GPT(D_o)$
Output: De-Identified dataset $D_d$
```



Prompt
Task: Please anonymize the following clinical note. **Task**
Specific Rules: Replace all the following information with the term ″[redacted]″: **Command**
Redact any strings that might be a name or acronym or initials, patients′ names, doctors′ names, the names of the M.D. or Dr., **NAME**
Redact any pager names, medical staff names, **NAME**
Redact any strings that might be a location or address, such as ″3970 Longview Drive″, **LOCATION**
Redact any strings that look like ″something years old″ or ″age 37″, **AGE**
Redact any dates and IDs and numbers and record dates, **ID-like strings**
Redact clinic and hospital names, **LOCATION**
Redact professions such as ″manager″, **PROFESSION**
Redact any contact information: **CONTACT**

Figure 6.11. This is the optimal prompt template we have designed for this task. We break down a prompt into 3 main segments: The task statement, the command, and specific rules. The task statement describes the goal and scope of the task. The command specifies actions to complete the task. Specific rules define specific requirements and optionally contain concrete examples. In the figure, we also annotate each line with a tag that describes its identity or correspondence in the PHI categories..

6.3.2.4 Designing Prompts to Incorporate HIPAA Guidelines

The HIPAA PHI categorization is the gold standard of defining clinical privacy. There are 18 HIPAA identifiers that are required to be de-identified, since this information can be used to identify, locate, or contact individuals. This is particularly important and relevant in processes (e.g, research and clinical collaborations) that involve data-sharing and transmission of clinical text documents. Figure **??** illustrates the mapping between HIPAA identifiers (see Table 6.7) and the i2b2/UTHealth benchmark used in this study. This correspondence relationship to HIPAA is generalizable (with proper changes) to any de-identification datasets.

Existing research clearly indicates that LLMs produce optimal results only when provided with the right prompts. To effectively utilize LLMs to redact sensitive information and ensure proper adherence to HIPAA guidelines, we design the following optimized prompts **template** for the benchmark dataset that unleashes the potential of LLMs.

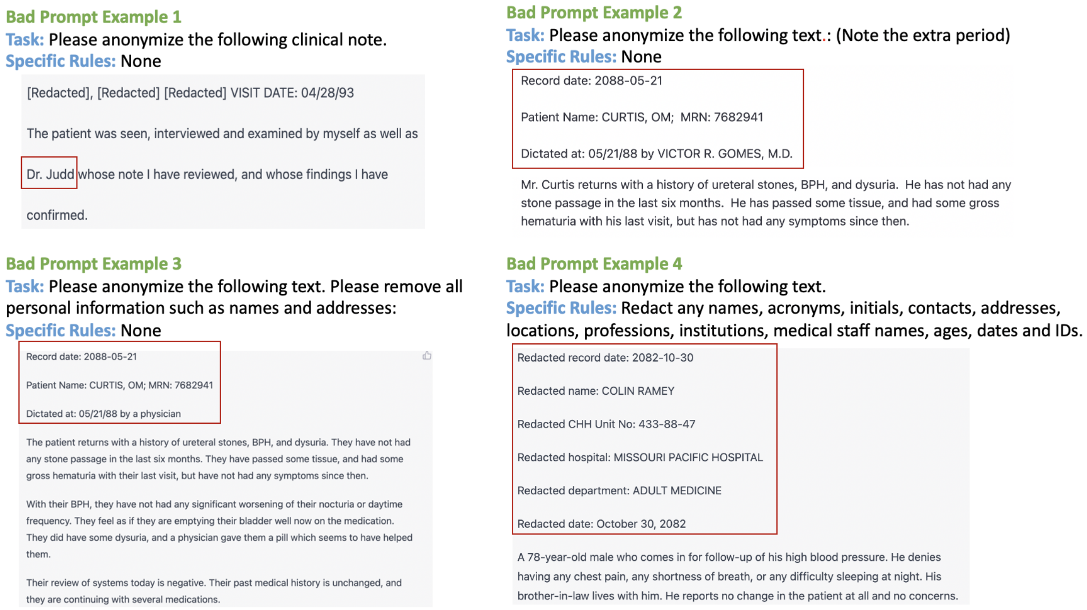
In this template, we first specify the **task** for the LLM to complete. It is necessary to provide the task information upfront, since it is easy for the user to comprehend and explain the task to be executed by the model. In addition, given the autoregressive nature (i.e., the next token is generated based on previous tokens) of the GPT-based models [187], it is important to push this information upfront for the LLM to better understand the user's demands and generate the most relevant results.

We then specify a special rule, (e.g., "Replace all the following information with the term "[redacted]":") that is associated with the declared task. This segment of the prompt can be adjusted to fit the specific task. We require the model to replace sensitive information with the token "[redacted]" for better **explainability**. This step also facilitates subsequent processing, including but not limited to result evaluation, surrogate information replacement (e.g., replacing the redacted information with fake synthetic data) and data-sharing.

Finally, we explicitly define the specific information that needs to be obscured. For example, we ask the model to "Redact any strings that might be a location or address, such as "3970 Longview Drive"". Ideally, the rules laid out in this segment should correspond to various categories of PHI in the target dataset. Based on our experience, these PHI categories are typically mapped to the HIPAA PHI guidelines, and it is necessary to cover all sensitive information so that the results are aligned with the HIPAA mandate as much as possible. Generally, it is helpful to explicitly specify such information and provide examples for better results and interpretability.

6.3.2.5 Prompting Caveats

Prompt design is an engineering process that combines art and science [252, 253]. Since this is the first study on utilizing LLMs to perform de-identification and the first work in exploring the ability to de-identify clinical data, we believe it is necessary to share our experience with both good examples and counterexamples.



Bad Prompt Example 1
Task: Please anonymize the following clinical note.
Specific Rules: None

[Redacted], [Redacted] [Redacted] VISIT DATE: 04/28/93

The patient was seen, interviewed and examined by myself as well as

Dr. Judd whose note I have reviewed, and whose findings I have

confirmed.

Bad Prompt Example 2
Task: Please anonymize the following text.: (Note the extra period)
Specific Rules: None

Record date: 2088-05-21

Patient Name: CURTIS, OM; MRN: 7682941

Dictated at: 05/21/88 by VICTOR R. GOMES, M.D.

Mr. Curtis returns with a history of ureteral stones, BPH, and dysuria. He has not had any stone passage in the last six months. He has passed some tissue, and had some gross hematuria with his last visit, but has not had any symptoms since then.

Bad Prompt Example 3
Task: Please anonymize the following text. Please remove all personal information such as names and addresses:
Specific Rules: None

Record date: 2088-05-21

Patient Name: CURTIS, OM; MRN: 7682941

Dictated at: 05/21/88 by a physician

The patient returns with a history of ureteral stones, BPH, and dysuria. They have not had any stone passage in the last six months. They have passed some tissue, and had some gross hematuria with their last visit, but have not had any symptoms since then.

With their BPH, they have not had any significant worsening of their nocturia or daytime frequency. They feel as if they are emptying their bladder well now on the medication. They did have some dysuria, and a physician gave them a pill which seems to have helped them.

Their review of systems today is negative. Their past medical history is unchanged, and they are continuing with several medications.

Bad Prompt Example 4
Task: Please anonymize the following text.
Specific Rules: Redact any names, acronyms, initials, contacts, addresses, locations, professions, institutions, medical staff names, ages, dates and IDs.

Redacted record date: 2082-10-30

Redacted name: COLIN RAMEY

Redacted CHH Unit No: 433-88-47

Redacted hospital: MISSOURI PACIFIC HOSPITAL

Redacted department: ADULT MEDICINE

Redacted date: October 30, 2082

A 78-year-old male who comes in for follow-up of his high blood pressure. He denies having any chest pain, any shortness of breath, or any difficulty sleeping at night. His brother-in-law lives with him. He reports no change in the patient at all and no concerns.

Figure 6.12. We present 4 examples of bad prompts. Example1: Only stating the task in the prompt is not optimal; Example2: Punctuation matters. The extra period made the command uninterpreted by ChatGPT; Example3: Multiple-task statement confuses ChatGPT; Example4: Did not explicitly specify the desired output.

We have presented the optimal prompt template in the previous section. In this section, we present ineffective prompts that lead to sub-optimal results. The list of ineffective prompts is non-exhaustive, and these samples are exemplary caveats.

6.3.3 Experiment & Results

6.3.3.1 Experimental Design

We compare ChatGPT and GPT-4 against several baselines methods, BERT [179], RoBERTa [240], ClinicalBERT [254].

BERT is the classic transformer-based language model [179]. The bi-directional transformer architecture enables impressive contextual understanding, and BERT has quickly become one of the most popular NLP models. It has been widely used in many domains and industries, and there are various offshoot models that are equipped with improved architectural design [255] or are trained on domain-specific data [256].

RoBERTa improves from BERT by removing the next sentence prediction (NSP) objective [240]. It is also trained with larger batches and on more training data. In addition, the masking patterns in RoBERTa are dynamically changed. Overall, RoBERTa can be regarded as a more refined version of the BERT model.

ClinicalBERT is a BERT-based model further pre-trained on the Medical Information Mart for Intensive Care III (MIMIC-III) dataset [229], a large collection of 2,083,180 clinical notes from the Beth Israel Deaconess Medical Center. This database contains data of 38,597 patients (admitted between 2001 and 2012) and is commonly used in clinical NLP [257]. The ClinicalBERT model is therefore a suitable model for medical text processing.

For all the baseline methods, we download pre-trained weights from Hugging Face [1] and initialize them on our local servers. BERT, RoBERTa and ClinicalBERT are initialized on Nvidia 3090 GPUs with 24 GB memories.

For all methods except GPT-4, we test all test cases in the test set of the i2b2/UTHealth data. However, since there is no public access to the GPT-4 API, one of our experts manually tested 50 random samples from the testing set to generate responses from GPT-4 using the OpenAI web interface. Regardless of the testing method, the generated responses go through the same evaluation script to calculate accuracy metrics.

6.3.3.2 Results

Experimental results show that GPT-4 achieves the highest de-identification accuracy (over 0.99) in a zero-shot scenario when provided with an optimal, explic-

---

[1] https://huggingface.co/

itly specified prompt. It outperforms the GPT-3 powered ChatGPT and all other baselines. Table 6.8 presents the complete experimental results.

Implicit Prompt

“Please anonymize the following clinical note” is an example of an implicit prompt. The desired outcome is stated, but the prompt lacks specific instructions and examples that help the LLM to carry out the task.

Explicit Prompt

Explicit prompts are prompts that contain concrete information that helps the LLM generate desired results within a clearly defined space. It contains a well-written and executable description of the desired output, clearly defines the task, and explicitly provides concrete examples. Please refer to Figure 6.11 for an example of a good explicit prompt.

It is noteworthy to point out that the BERT-based baselines are fine-tuned on the dataset through a rigorous supervised learning named entity recognition (NER) process. Despite their strong performance (both RoBERTa and ClinicalBERT attain over 90% accuracy rates), the fine-tuning process requires significantly more time and engineering efforts to complete, compared to the streamlined process offered by large language models such as ChatGPT and GPT-4. Our clinician experts believe this demonstrates the attractiveness of employing large language models over standard-sized models for the de-identification task, since LLMs require significantly less efforts and are accessible to health providers who have limited experience in machine learning.

In addition, we note that carefully crafted prompts can significantly improve LLM performance. For example, an optimally designed prompt improves ChatGPT performance from 0.686 to 0.929. GPT-4 is less susceptible to defects in the prompt,

Table 6.8. Clinical notes de-identification results with LLMs.

De-Identification	i2b2			
	Implicit Prompt* (Zero-shot)	Explicit Prompt* (Zero-shot)	General Prompt* (Zero-shot)	Fine-tuning
ChatGPT	0.686	0.929	N/Aw	N/A
GPT-4	**0.908**	**0.99**	N/A	N/A
BERT	-	-	-	0.798
RoBERTa	-	-	-	0.947
ClinicalBERT	-	-	-	0.974
mT0	0.824	N/A	0.827	N/A
Falcon-7b	0.603	0.605	0.597	N/A
Flan-t5-base	N/A	N/A	0.737	N/A
Llama1-7b	0.609	0.612	0.597	N/A
Llama2-7b	0.609	0.612	0.597	N/A

*Note: The prompting based methods are essentially performing Zero-Shot learning

since it nonetheless performs well even when fed with a simple, implicit prompt. However, an optimally designed prompt propels the accuracy of GPT-4 to a new level that is readily usable even for real-world applications.

6.3.3.3 De-Identification Quality Evaluation

The performance of the de-identification methods is evaluated through entity-wise accuracy, defined by the percentage of sensitive entities removed after the inputs are processed by the corresponding method.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.2}$$

where TP, TN, FP, and FN denote numbers of true positives, true negatives, false positives, and false negatives, respectively.

6.3.3.4 Error Analysis

We conduct an error analysis by examining the incorrect answers generated by various LLMs. We have identified significant differences in the nature of errors made by these models compared to GPT-4 and ChatGPT. The errors observed in GPT-4 and ChatGPT are predominantly associated with their limitations in recognizing relevant entities, which can be attributed to their proficiency and experience in this specific task. On the other hand, the errors in the Llama series and Falcon can be primarily attributed to a fundamental misunderstanding of the task itself, often resulting in nonsensical responses, such as merely repeating the content of the given prompt. Additionally, an excessively lengthy prompt can also lead to LLMs struggling to grasp its meaning. Overall, we conclude that only a small subset of these LLMs demonstrate the potential for swift practical application, and the majority of LLMs still have significant room for improvement.

6.3.4 Discussion and Conclusion

6.3.4.1 Locally-Deployed DeID-GPT for Hospital Use

Despite the impressive performance of ChatGPT and GPT-4 for data de-identification, these models could only be accessed through online APIs, making them impossible to be applied in a hospital setting as patient data cannot be stored nor transmitted to a non-authorized external party. Furthermore, as the code and implementation details of ChatGPT/GPT-4 are not open to the public, we cannot fully validate their functionalities to ensure these models are HIPAA-compliant and meet the standards of the hospital's quality management system (QMS). Therefore, it would be vital to use open-source or in-house trained LLMs for the de-identification task and deploy them locally to ensure data security, privacy, integrity, and proper adherence to HIPAA guidelines. Thus, we are investigating solutions for the locally-deployed DeID-GPT. For example, we experimented with LLaMA [234] in this work to preliminarily examine its performance. However, LLaMA failed to generate coherent and relevant outputs for all test cases, and is therefore not directly comparable to other tested methods. We also intend to try other open-source LLMs such as OPT [258] and BLOOM [239]. These models are open-sourced and were trained with data on the public domain, making it possible to reuse without licensing concerns [234]. To deal with the challenge of limited computational resources at the local site, we will also explore the state-of-the-art model quantization [259], acceleration [260] and GPU parallelization [261, 262] techniques to streamline the deployment.

6.3.4.2 Developing Domain-specific LLMs

To the best of our knowledge, there are no LLMs specifically tailored to the medicine and healthcare domain with medical text such as the clinical notes and radiology reports used in this work, let alone for more diversified medical specialties. Given the success of previous domain-specific language models such as BioBERT [241] and BioGPT [263], we believe it is promising and helpful to develop domain-specific LLMs. At the data level, it is possible to continue the pre-training process on domain-specific data. This is a proven route to success since it is well known that language models can perform better on domain-specific benchmarks and tasks when trained with relevant data and knowledge [256, 264] as the models are exposed to domain vocabulary, jargons, terms and writing styles. At the architecture level, it is also possible to modify and optimize LLMs for specific domains. For example, it is possible to design modules that can more effectively capture sensitive HIPAA-protected information by prioritizing such information in the input [265]. It might also be insightful to explore the possibility of efficient passing of inputs with sparsely activated forward pass blocks [266] to achieve performance or efficiency gains.

6.3.4.3 Improving De-Identification with LLMs through Fine-tuning

The upcoming public release of GPT-4's API service will enable fine-tuning of the model. Given the impressive performance of GPT-4 in the zero-shot prompting setting, we expect that it can deliver even better performance with fine-tuning to the domain-specific data (e.g., clinical notes and reports). On the other hand, to better ensure data privacy and security, it is also necessary to investigate the potential of replicating GPT-4-like performance locally with other LLMs. On local servers, it is more necessary and beneficial to evaluate the impact of fine-tuning on task performance, as locally-deployable models generally have a smaller parameter size. Regardless of the approaches, we see fine-tuning LLMs as a promising research direction that enables the practical use of LLMs in a hospital setting.

6.3.4.4 Applying LLM to Anonymize Other Data

Text de-identification and data anonymization are crucial in many other scenarios where personal information needs to be collected or shared and privacy protection is paramount. While our approach focuses on medical data, which requires particularly stringent privacy protections, it could be extended to other sensitive data domains beyond healthcare. For example, financial institutions, such as banks and insurance providers, may need to de-identify data containing customer information, such as account numbers and transaction details, to safeguard individuals' privacy. Market research companies also collect data on individuals' opinions, behaviors, and preferences, which should be de-identified to protect the privacy of research participants. Additionally, sensitive information may need to be shared in legal proceedings, including financial records, medical records, and personal information that needs to be de-identified to preserve individuals' privacy. Our proposed ChatGPT/GPT-4-

based approach in this study could potentially be generalized to these scenarios in the future, offering a novel approach to data de-identification by utilizing LLMs.

6.3.5 Future Perspective

As far as we know, this study is the first work in the NLP literature to investigate the possibility of employing LLMs such as ChatGPT and GPT-4 for data de-identification. In particular, we partnered with clinicians to evaluate these powerfully versatile modern models on medical text anonymization. Experimental results indicate that ChatGPT and GPT-4 have excellent abilities in de-identifying medical data compared to other LLMs. The application of LLMs to medical text data has already shown promise in providing valuable insights into various medical conditions and diseases, and there is significant potential for future integration with other modalities such as medical imaging data [267].

With its multi-modal capabilities, GPT-4 can be explored for the integration of multimodal medical data such as radiological images, pathological images, clinical text reports, and genomics data, among others, to gain new insights for disease diagnoses, treatments, follow-up, and prognosis. By enabling cross-modality analysis using GPT-4 and similar methodologies, we can significantly enhance our understanding of different medical conditions, including various types of cancers, brain disorders, cardiovascular diseases, and many other human diseases. Ultimately, the development and application of GPT-4 related approaches for multi-modal medical data have great potential to revolutionize healthcare.

Reciprocally, healthcare professionals can also contribute to the advancement of LLMs such as GPT-4 and its related models/methods through their domain expertise and expert feedback. For instance, highly skilled and professional medical physicians can advance the reinforcement learning by human feedback (RLHF) used in ChatGPT/GPT-4 to reinforcement learning by expert feedback (RLEF) in the medical domain. We envision that this RLEF framework can significantly speed up the pace of widely adopting GPT-4 and its future variants in the medical and healthcare field.

CHAPTER 7

A UNIFIED AND BIOLOGICALLY-PLAUSIBLE RELATIONAL GRAPH REPRESENTATION OF VISION TRANSFORMERS

Improving the efficiency of artificial neural networks (ANN) is an important problem in AI domain. Various research works have focused on discovering optimal structures for more efficient ANNs. A representative one is the “Relational Graph”, published at ICML in 2020. In this study, the authors introduced a relational graph method that allows the ANNs to be represented as a graph, and conversely, given a graph, it can generate the corresponding ANN. By employing this approach, the search for an optimal ANN structure can be conducted by systematically examining different graphs within a graph space system. Through extensive experiments, it was observed that the graph structures of the top-performing ANNs surprisingly similar to those of real biological neural networks (BNNs). But this study only explored the Multi-Layer Perceptron (MLP) and CNN-based models. Whether the latest transformer architecture in the field of AI have the same conclusions is unknown. To address this gap, in **Chapter 7**, we extended the evaluation to include the Transformer architecture. By this way, the three primary structures inclusing – MLP, CNN, and Transformer, can all be represented by graph in a unified graph space. We compared graph structures of best-performing MLP, CNN-based models and transformer-based models with different BNNs. The result might suggest that models similar with the more advanced biological brain networks perform better.

7.1 Introduction

The human brain is a hugely complex, highly recurrent, robust, and remarkably efficient nonlinear neural network [268]. Human behavior, cognition, and neural activities critically depend on the graph structure of large-scale brain connectome [269, 270]. Likewise, representation and predictive capability of an artificial neural network (ANN), such as convolutional neural network (CNN), are also closely related to the model's graph structure [271, 12]. The authors in [12] hypothesized that biological neural networks (BNNs) (e.g., macaque's brain network graphs) and ANNs (CNN's relational graphs) might share common graph properties. Recently, neural networks of transformers have received extensive attention, and a variety of vision transformers (ViTs) have been developed rapidly in the computer vision field. Tolstikhin et al. [272] and Yu et al. [273] summarized the message passing in transformers as Token Mixer and Channel MLP (multilayer perceptron). The core idea of various ViT models is the well-designed Token Mixer which controls the information communication between spatial tokens, while most ViTs follow similar design of the Channel MLP (two fully-connected linear with GELU function), for examples, the self-attention [274] in ViT [5], the shifted window attention in Swin [275], the token-mixing MLP in Mixer [272, 276], the average pooling in PoolFormer [273], and so on. Essentially, exploring ViT's graph structure (in terms of information communication), its relevance to the representation/predictive performance, and its similarity to BNNs, is of great importance for deeper understanding of ViTs and their wider applications.

The work in this paper is inspired by the relational graph constructed on CNN models [12]. You and colleagues [12] introduced the pioneering concept of *relational graph* which considers channels of CNN features as nodes. They systematically investigated how the relational graph structure of CNN networks affects their predictive performance. They found sweet spots in the relational graphs of CNN networks, which are similar to the real BNNs reconstructed from brain science data. Built upon the inspiring work in [12], in this paper, we proposed the following major innovations. (1) The relational graph in [12] was constructed only on the multilayer perceptron (MLP) or CNNs. Here, we will focus on defining and constructing relational graphs on ViT models. (2) The relational graph in [12] only considered the communication between the dimensions or channels of CNN, thus ignoring the spatial information aggregation. Here, we will consider the spatial patch information communication in ViT as a key factor. (3) The relational graph in [12] only described the network's graph topology, thus ignoring the influence of parameter weights which can reflect the ability of capturing embedding features from images. Here, we will consider the influence of parameter weights in ViT as a key factor. Overall, our work provides a novel and unified relational graph representation of different ViT architectures for systematic understanding and assessment of model representation performance.

Specifically, inspired by PoolFormer and MLP-Mixer [273, 272], we divided the *relational graph* in ViT models into *aggregation graph* and *affine graph*, where the *aggregation graph* describes the spatial patch information interaction, while the *affine graph* reflects the information communication between channels. For example, in ViT transformer, we considered the self-attention as an aggregation network and the feedforward network (FFN) as an affine network. In this work, we explored five types of widely known ViT models, including ViT [5], DeiT[277], Swin [275], Mixer [272] and PoolFormer [273] as examples.

In order to systematically explore the advantages of ViT models, we investigated the relationships between two sets of key factors. (1) ViT's relational graph structure vs its predictive performance. To relate to the model parameters rather than only model's graph structure, especially when pre-training/fine-tuning has become a popular methodology for various vision tasks, more attention was paid to the feature extraction ability of the ViT model's backbone. We evaluated the classification performances of these different models with fixed backbones and graph structures. (2) ViT's relational graph structure vs those of BNNs. The convolutional and pooling layers in CNNs were directly inspired by the classic notions of simple cells and complex cells in visual neuroscience [33, 278]. Here, we explored the similarities of spatial information communication patterns represented by the proposed relational graphs in ViT models and those in real BNNs represented by brain graphs derived from neuroscience data. Finally, based on our constructed aggregation graph, we propose a relational graph loss for further training and improving the model performance.

7.2 Relational Graph in Vision Transformers

7.2.1 Structure of Vision Transformer Models

As shown in Fig. 7.1(c), there are three core parts in vision transformers: Patch Embedding Fig. 7.1(a), Token Mixe Fig. 7.1(b) and Channels MLP Fig. 7.1(d). The Patch Embedding layer projects the input images into non-overlapping image tokens as word tokens in NLP. The Token Mixer allows information communication between different spatial locations (images tokens) while the Channels MLP allows information communication between different channels. Therefore, the overall structure of vision transformers is represented as:

$$X = PatchEmbed(Image) \tag{7.1}$$

$$Y = TokenMixer(Norm(X)) + X \tag{7.2}$$

$$Z = ChannelsMLP(Norm(Y)) + Y \tag{7.3}$$

where $Norm$ is the normalization function. Note that there are also some other model designs such as the learnable absolute position embedding and class token in ViT, the relative position bias in Swin, the distillation token in DeiT, etc. Here, we only discuss the backbone of the model to simplify the issue.

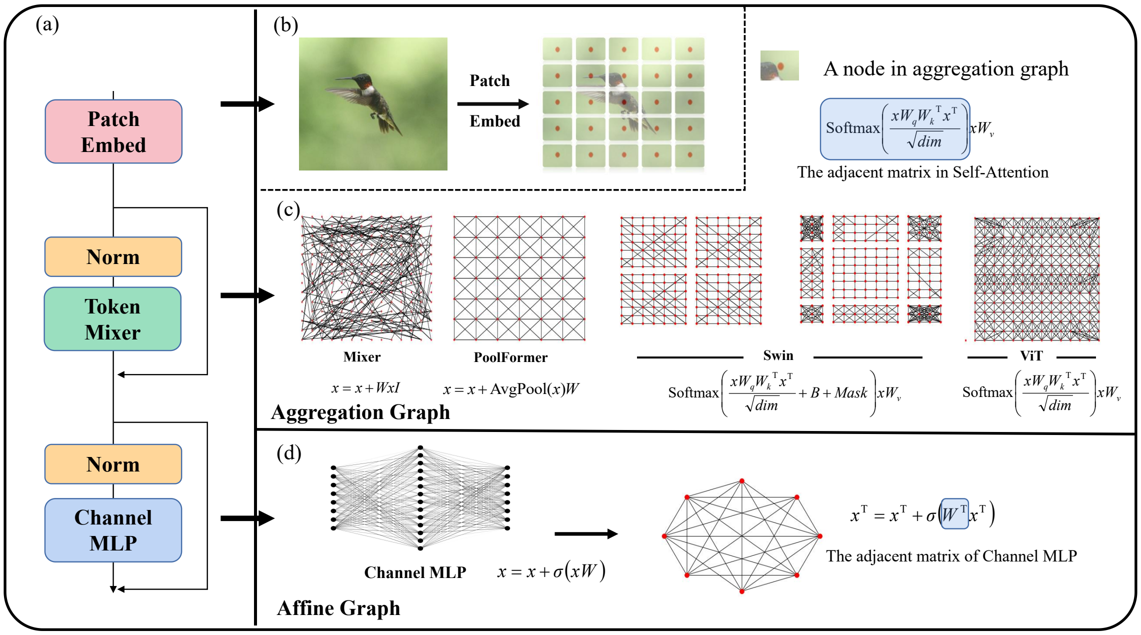

(a)
Patch Embed
Norm
Token Mixer
Norm
Channel MLP
(b)
Patch
Embed
A node in aggregation graph
$\text{Softmax}\left(\frac{xW_qW_k^\top x^\top}{\sqrt{dim}}\right)xW_v$
The adjacent matrix in Self-Attention
(c)
Mixer
PoolFormer
Swin
ViT
$x = x + WxI$
$x = x + \text{AvgPool}(x)W$
$\text{Softmax}\left(\frac{xW_qW_k^\top x^\top}{\sqrt{dim}} + B + Mask\right)xW_v$
$\text{Softmax}\left(\frac{xW_qW_k^\top x^\top}{\sqrt{dim}}\right)xW_v$
Aggregation Graph
(d)
Channel MLP
$x = x + \sigma(xW)$
$x^\top = x^\top + \sigma(W^\top x^\top)$
The adjacent matrix of Channel MLP
Affine Graph

Figure 7.1. The aggregation and affine graph in different ViT models. (a) The patch embedding in vision transformer models. (b) The aggregation graph and (d) the affine graph in vision transformer models. (c) the structure of transformer vision models.

7.2.2 Aggregation and Affine Graphs

Inspired by the PoolFormer model [273], the two core message exchange functions of vision transformer are Token Mixer and Channel MLP. The Token Mixer aggregates information from tokens while the Channel MLP allows information communication between different channels [272]. Therefore, we defined the relational graph of vision transformer as two key sub-graphs: Token Mixer Graph and Channel

Mixer Graph, and formally annotated them as Aggregation Graph and Affine Graph according to their major roles.

Aggregation Graph By considering each token as a node in the graph, the token mixer (e.g. self-attention) plays the same role as the aggregation function in graph neural networks via updating their embedding features by aggregation of the information from its neighborhoods. With this message-passing paradigm, the ViT model can globally integrate information from different regions (tokens) in each layer, whereas for CNN models, message passing is restricted to the kernel size. For example, the propagation function between tokens of MLP-Mixer is written as:

$$x = x + (LN(x^T)W)^T (7.4)$$

where LN is the LayerNorm function and W is the set of learning parameters. If we regard each token as a node in graph, this function is written as $TokenMixer(x) = W^T xI$. Therefore, the normalized adjacent matrix of aggregation graph of Mixer is $Softmax(W^T/\sqrt{dim})$. Note that we used the same normalization function as [5, 274], which can also be written as $\hat{A} = AD^{-1}$ where $A = exp(W^T/\sqrt{dim})$ and $D_{ii} = \sum_j A_{ij}$).

Affine Graph By considering each channel as a node in the graph, the channel mixer allows the message passing along channels in the affine graph. Different from the aggregation graph which conveys spatial information of tokens, the affine graph is designed for the information communication between different channels. The channel mixer always contains two full-connection layers and a nonlinear activation function as:

$$x = x + \delta(LN(x)W_1)W_2 (7.5)$$

where δ is the nonlinear activation function such as GELU and ReLU. Therefore, the adjacent matrix of affine graph is written as $Softmax(W_1 W_2/\sqrt{dim})$. We add the

$Softmax$ function to norm the matrix and make the connection matrix non-negative, in alignment with the aggregation graph.

7.2.3 Construction of Aggregation Relational Graph in Vision Transformers

Since the key difference among different ViT models is the different design of Token Mixer, i.e., aggregation graph, we therefore mainly focused on a unified aggregation relational graph representation of different ViT models, instead of an affine one in this study. Specifically, we selected ViT[5], DeiT[277], Swin[275], Mixer[272] and PoolFormer[273] as five representative examples among all ViT models as introduced in Related Work **??** due to the space limit.

ViT/DeiT The ViT [5] and DeiT [277] models both follow a pure transformer [274] design while more training schemes are adopted in DeiT. Both ViT and DeiT adopt the multi-head self-attention (MHSA) as the Token Mixer, which can be described as:

$$Softmax(xW_qW_k^Tx^T/\sqrt{dim})xW_v(7.6)$$

In addition, they keep the position information with learnable absolute position embedding $x = x + P$ where P is the position embedding. For the sake of analysis, we assumed that different tokens are independent of each other, and the information communication between tokens is carried out through position embedding. Therefore, the adjacent matrix of aggregation graph is written as:

$$\hat{A} = Softmax(PW_qW_k^TP^T/\sqrt{dim})(7.7)$$

It is worth noting that we did not use the information of the W_v matrix due to its role as a feature projection rather than an exchange of spatial information between tokens.

Swin Compared with ViT, Swin [275] proposes the windows attention and shifted windows attention in order to introduce inductive bias into vision transformer. Additionally, relative position bias rather than absolute position embedding is applied in Swin. The Token Mixer in windows of Swin is:

$$Softmax(xW_qW_k^Tx^T/\sqrt{dim} + B + Mask)xW_v (7.8)$$

where B is the relative position bias and $Mask$ is designed for shifted attention operation, which is window-specific. Therefore, the aggregation graph in each window is:

$$\hat{A} = Softmax(I/\sqrt{dim} + B + Mask)(7.9)$$

where I is the unit matrix.

Mixer The MLP-Mixer is an MLP only vision model [272, 276], primarily composed of token-mixing MLP and channel-mixing MLP. The token-mixing MLP acts on the transpose matrix of the image token, thus aggregating spatial information between different tokens while the channel-mixing MLP is the same as the FFN in transformer. Note that there is no position embedding in Mixer as the token-mixing MLPs are sensitive to the order of the input tokens. Based on the token mixer defined in (7.4), the aggregation graph is written as:

$$\hat{A} = Softmax(W/\sqrt{dim})(7.10)$$

where W is the set of learning parameters of MixingToken in (7.4).

PoolFormer Based on the MLP-Mixer, Yu et al. [273] proposed a PoolFormer which uses pooling functions to perform information communication between tokens and greatly reduces the computational effort and parameters of the model. Therefore, the aggregation graph of PoolFormer is:

$$a_{i,j} = 1/K^2, j \in Ner(i)(7.11)$$

where K is the kernel size and $Ner(*)$ is the neighborhood set.

7.2.4 Aggregation Graph between Layers

The considerable difference in depth and width of different ViT models makes it challenging to adopt a unified framework for model assessment. You [12] assumed multiple features as one node in order and layers as rounds, thus making different models comparable. Inspired by this study, we fixed the aggregation graph size as $14 \times 14(+1)$ nodes and also regarded the layers as rounds. Therefore, for the high resolution aggregation graph, we down-sampled it by:

$$Downsample(\hat{A}_{xy}) = 1/K * \sum_{i}^{Ner(x)} \sum_{j}^{Ner(y)} \hat{A}_{ij}(7.12)$$

where K is the down-sampling rate and $Ner(*)$ is the neighborhood set in the high resolution image. For the low resolution aggregation graph, we up-sampled it by:

$$Upsample(\hat{A}_{xy}) = 1/K\hat{A}_{ij}, i = x//K, j = y//K(7.13)$$

where i, j is the index of the raw resolution image and x, y is the index of downsampled or upsampled image. The final aggregation graph of a model is:

$$Final\hat{A} = \prod^{Layers} Sampled(\hat{A})(7.14)$$

Note that we normalized $(Softmax(Final\hat{A}))$ matrix for ease of analysis.

7.3 Experiments

Following the suggestion in [12], we adopted two graph measures, the average path length and clustering coefficient that characterize the integration and segregation of a network respectively, to systematically explore (1) the relationship between these graph measures and downstream task performances of ViT models, (2) the ef-

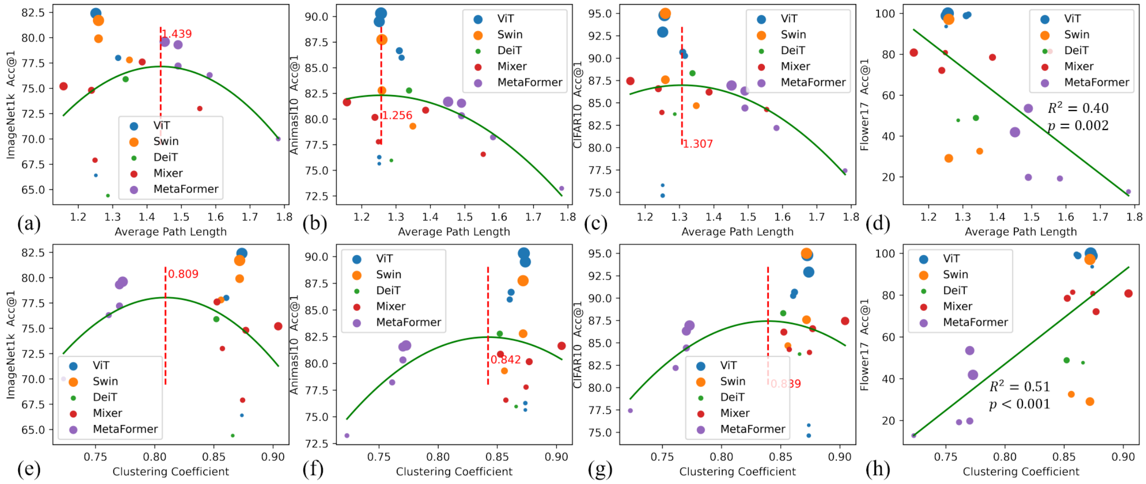


Figure 7.2. Relationship between two aggregation graph measures and classification performance of 21 ViT models on four datasets, Larger spot size represents larger model parameters..

fectiveness of these graph measures as ViT model training indicators from scratch, and (3) the topological similarity between ViT models and real BNNs.

7.3.1 Downstream Task of Pretrained ViT Models

We used image classification as the downstream task and evaluated the performance of the model's backbone on four different and widely-known datasets: ImageNet-1k [279] including 1000 categories of images with more than 1.28 million of images, CIFAR10 [280] including 50K training images and 10K validation images, Animal10 [281] including 50k training images and 5k validation images, and Flower17 [282] including 80 images (60 for training, 10 for testing and 10 for validation) for each category.

Models We adopted 21 publicly available pretrained models of the five ViTs (ViT, DeiT, Swin, Mixer, and PoolFormer) provided in [283], including vit-tiny-patch16-224[5], vit-tiny-patch16-224-in21k[5], vit-small-patch16-224[5], vit-small-patch16-

224-in21k[5], vit-base-patch16-224[5], vit-base-patch16-224-in21k[5], swin-tiny-path4-window7-224[275], swin-small-path4-window7-224[275], swin-base-path4-window7-224[275], deit-tiny-patch16-224[277], deit-small-patch16-224[277], resmlp-12-224[276], resmlp-12-distilled-224[276], resmlp-24-224[276], resmlp-24-distilled-224[276], resmlp-36-224[276], poolformer-s12[273], poolformer-s24[273], poolformer-s36[273], poolformer-m36[273], poolformer-m48[273]. The pretrained vision models were all downloaded from timm [1].

Settings We followed the same setting of [284] for better model training[2]. An SGD optimizer with a weight decay of 0.0001 was applied. We employed an OneCycleLR [285] learning rate scheduler and the initial(max) learning rate was 0.0005(0.001) with cosine annealing. The percentage of the cycle spent increasing the learning rate was set to 0.1. The label smoothing of CrossEntropy Loss was set to 0.2 and batch size was 128/256. The data augmentations included Cutout [286] and RandAugment [287]. All images were resized to 224 × 224 in consistent with the pretrained model. All models were with pretrained parameter weights and only the parameters of classification header would change while the backbone was fixed. For the ImageNet-1k dataset, no parameter was learned and we just tested on the validation dataset as they have been pre-trained on the ImageNet-1k. Three ViTs which are pre-trained on the different ImageNet-21k were discarded to keep the consistency of pre-trained model.

7.3.2 ViT Model Training from Scratch

To explore the dynamic changing characteristics of the two graph measures during model training and their effectiveness as ViT model training indicators, we

---

[1] https://github.com/rwightman/pytorch-image-models

[2] https://github.com/Alibaba-MIIL/ImageNet21K

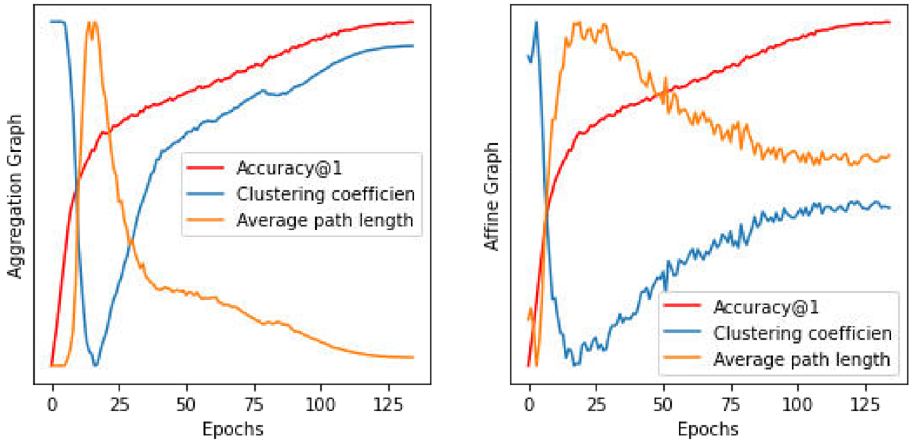


Figure 7.3. Dynamic change of the two graph measures during ViT-Ti model training from scratch on ImageNet-1k dataset..

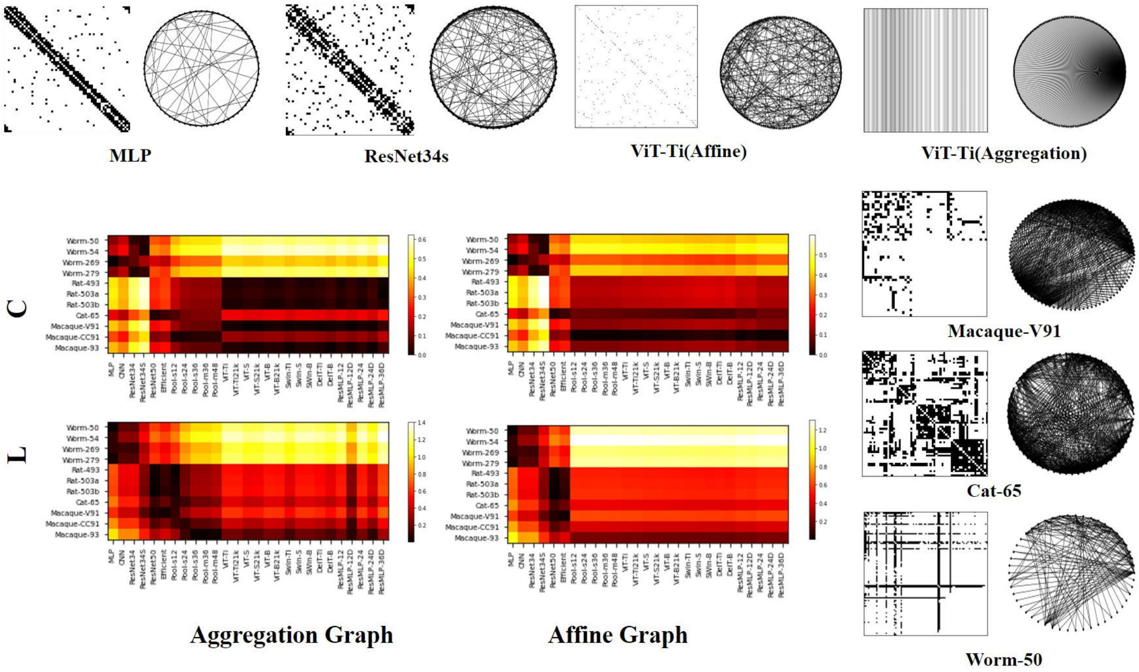


Figure 7.4. Similarity of relational graphs between vision models and BNNs. L: the average path length, C: the clustering coefficient..

trained the ViT-Ti [5] from scratch on ImageNet-1k [288] using the same setting as in 7.3.1 and recorded the accuracy and graph measures for each epoch. We used Adam optimizer with batch size of 800 for better performance.

7.3.3 Graph Similarities between ViTs and BNNs

A variety of real BNNs were compared to the ViT relational graphs, including the synaptic connectomes of the anterior (Wrom_279) and posterior (Worm_269) nervous system of the C. elegans; the synaptic connectomes for the pharyngeal nervous systems of two nematodes with divergent feeding behavior (Worm_54 and Worm_50); the neuronal connectomes of the rat brain revealed by neuronal pathway tracers (Rat_493, Rat_503a and Rat_503b); the macroscopic brain network of cat as reconstructed from tract tracing data (Cat_65); the structural connectome of a macaque monkey derived from a collation of tract tracing studies (Macaque_242); the interareal connectivities of macaque monkeys (Macaque_V91, Macaque_CC91 and Macaque_93) revealed by retrograde tracers. We adopted the publicly available BNNs data with approval, and a detailed description of those BNNs is referred to [3]. The Euclidean distance between the graph measures of BNNs and ViTs was calculated as the topological similarity between BNNs and ViTs.

7.4 Result

7.4.1 Relationship between Aggregation Graph Measures and Classification Performances

There was a smooth U-shape correlation between the model classification performance and each of the two graph measures in the first three datasets using a second-degree polynomial regression as illustrated in Fig. 7.2(a,b,c,e,f,g). Sim-

---

[3] https://neurodata.io/project/connectomes/

ilarly in [12], we successfully identified stable sweet spots $C \in [0.809, 0.842]$ and $L \in [1.256, 1.439]$ across the three different datasets as highlighted as red dashed lines in Fig. 7.2(a,b,c,e,f,g) corresponding to the best classification performance.

As illustrated in Fig. 7.2(d,h), the relationship between the graph measure and classification performance in the fourth dataset (Flower17) exhibited a significant linear correlation instead of a U-shape one. The interpretation was that the Flower17 dataset has only 60 images per category similar to few-shot learning task. Therefore, these two graph measures of the proposed relational graph not only helped us identify the sweet spot with large sample data, but also reflected the adaptive learning ability of the model with few sample data. In summary, our results illustrated that the two graph measures can be used as effective indicators of model classification performance.

7.4.2 Dynamic Changing Characteristics of Graph Measures during Model Training

We averaged the graph measures of each layer as the representation of the final graph. A threshold of 1/192 (embedding dim/node numbers) for the affine graph and 1/197 (token numbers) for the aggregation graph was selected to illustrate the dynamic change of graph structure during model training. Note that we concatenated all aggregation graphs along the diagonal rather than through rounds in order to better highlight the changing trend of the model. We reported the top-1 accuracy on validation dataset and two graph measures in Fig. 7.3. Interestingly, except for the initial 25 epochs, there was considerable consistency between the graph measure and the accuracy of the model, indicating that the two graph measures may also serve as a more general and effective indicator of model training.

7.4.3 Graph Measure Similarity between ViTs and BNNs

The graph measure similarity between ViT models and BNNs is shown in Fig. 7.4. Note that the relational graphs of MLP and CNN based models are from [12]. We see that the graph measures of ViTs were close to the rat, cat and macaque's brain neural networks, while those of the MLP and CNN based models are similar to worms. This inspiring result might suggest the superiority of ViT over MLP/CNN in terms of information communication and exchange efficiency, given that mammalian brains such as the rat, cat and macaque are considered to be much more advanced and optimized than the worm's neural networks.

7.4.4 Aggregation Graph on Model Design

The result of different models with different aggregation graphs on image classification demonstrates that the closer the model is to the sweet spot, the higher performance of the model. Therefore, it inspires us to optimize the model design for a better performance via guiding the target model and the reference model to have a similar aggregation relational graph. To achieve a closeness between the relational graphs of the target model and the reference model, we define the similarity loss of the relational graph as:

$$\mathcal{L}_{(\theta_t,\theta_r)} = \sum^{Layer} |A_t - A_r| (7.15)$$

where θ_t, θ_r are the parameters of the target model and reference model, A_t, A_r are the aggregation graph defined in Section 7.2.3. Therefore, the training of the target model could be written as Algorithm 1. Note that we do not use any additional image data for training, which results in a short training time (several minutes in this study).

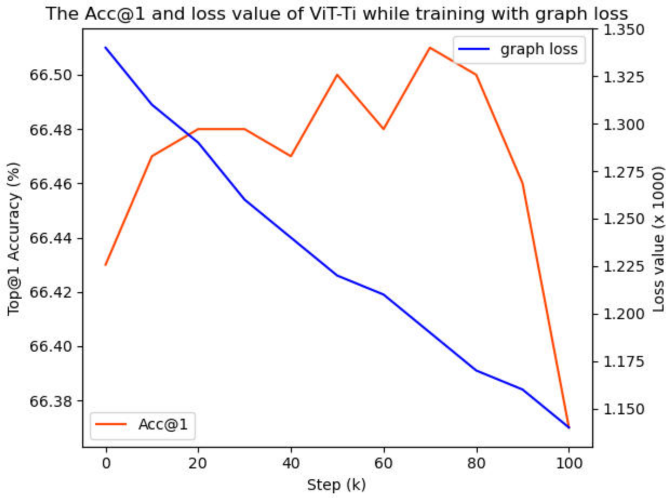


Figure 7.5. The accuracy and graph loss value during training when the ImageNet-1k pretrained ViT-T was as target model and the ImageNet-21k pretrained ViT-B was as reference model..

Algorithm 1 Algorithm of training with relational graph loss

Require: Reference model with parameters θ_r and Target model with θ_t.

```
Initialize learning rate $\eta$;
while Training do
    $Loss \leftarrow \mathcal{L}_(\theta_t, \theta_r)$ in (7.15)
    $\theta_t \leftarrow \theta_t - \eta \nabla L(\theta_t)$
end while
```

We evaluated the algorithm on ViTs and reported the top 1 accuracy on ImageNet-1k validation dataset. Fig. 7.5 shows the accuracy and loss value of ViT-T model when training with ViT-B as the reference model. During the model training, the accuracy of the model improved from an initial 66.43% to a maximum of 66.51% and finally decreased to 66.37%. The results showed that the aggregation graph and the model accuracy were closely related, and higher similarity with better model would improve the accuracy of the model. However, as the model was further trained, the model accuracy began to decrease, which might be attributed to the lack of classification loss constraints on the model parameters. An effective suggestion is to stop training once the graph loss is partially reduced, e.g. 5%.

To verify the applicability of graph loss, we used the ImageNet-1k pre-trained models as the target models and the ImageNet-21k and ImageNet-1k pre-trained models as the reference models, respectively. The results in Fig. 7.6 illustrate that when using a superior model as the reference model, the accuracy of the model will be improved after a short period of training. Conversely, when using a less superior model as the reference model, the accuracy of the model will be reduced, which proves the effectiveness of our method, especially since the model does not use any image data.

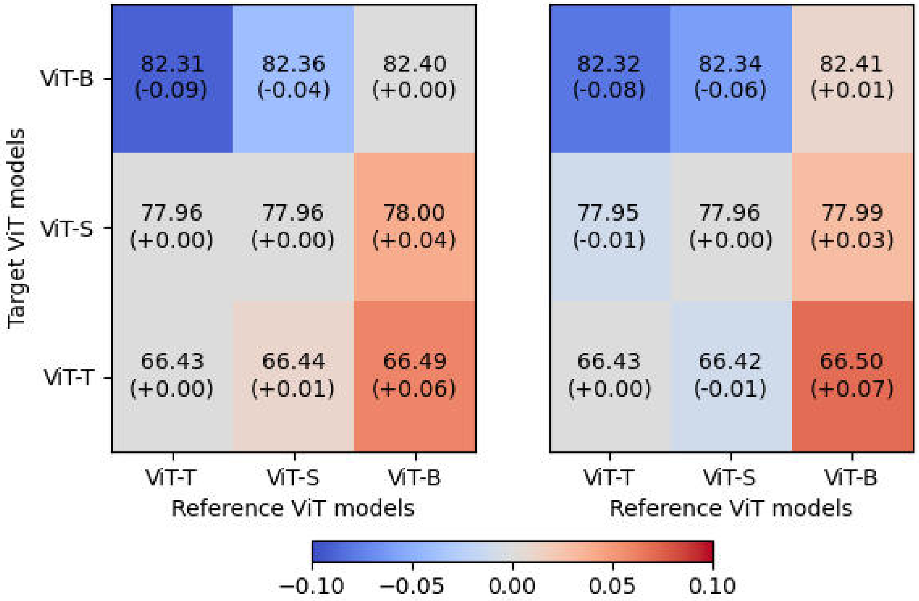


Figure 7.6. The accuracy changed on ImageNet-1k after ViT models trained with graph loss. (left) The reference model also pretrained on ImageNet-1k. (right) The reference model pretrained on ImageNet-21k..

7.5 Discussion

7.5.1 Fine-tune Partial vs. Full Model

The construction of the proposed relational graph was affected by ViT model parameters. To keep consistent with the graph structure, we froze the parameters of ViT model's backbone and only updated the classification header. To justify weather fixed backbone would affect the assessment of model performance, we provided the predictive performance on Flower17 dataset with/without freezing the backbone parameters in Fig. 7.7. We observed that the performance with/without freezing has a high linear correlation, indicating that it is reasonable to assess the model performance by graph measures with fixed backbone.

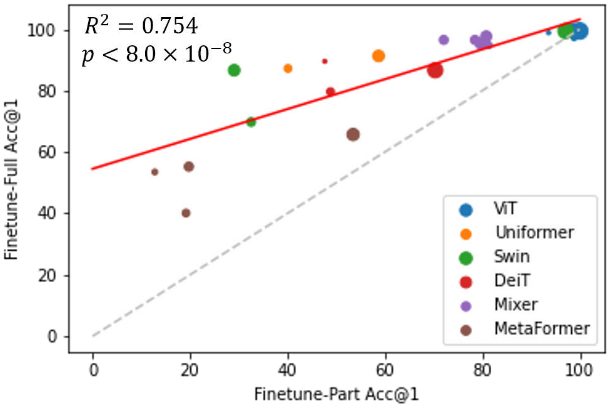


Figure 7.7. The prediction accuracy on Flower17 with fine-tuning partial and full models.

7.5.2 Graph Structures with Different Sampling Efficiencies

Section 7.4.2 demonstrated that the graph measures changed during model training with the same training setting. We further reported the graph measure changes of ViT with different training strategies such as training with more datasets

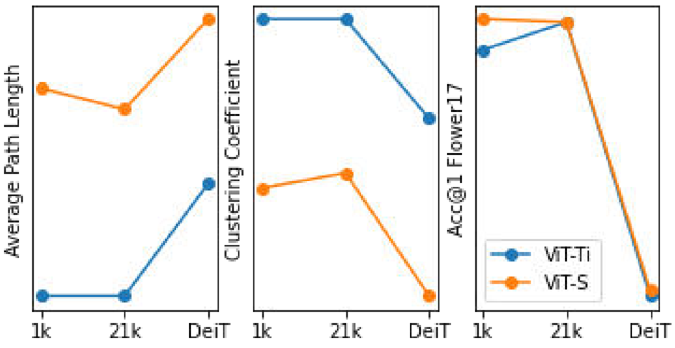


Figure 7.8. The graph measures of different training methods in the same model.

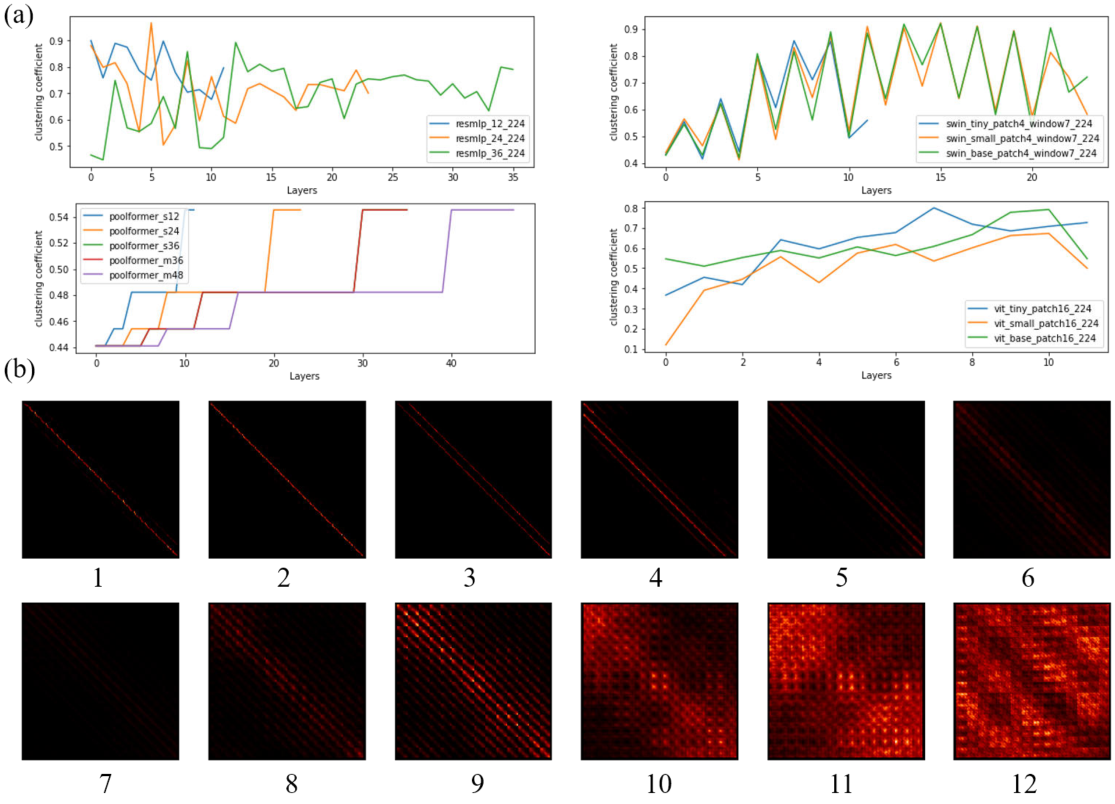


Figure 7.9. The aggregation graphs between layers in different ViT models. (a) The clustering coefficients of Mixer, MetaFormer, ViT and Swin. (b) The visualization of aggregation graphs in ViT-B..

or with distillation learning in Figure 6. Compared with DeiT with more training schemes [277], the graph measures of models pretrained with more data (ImageNet-21k pretrained) had higher similarity with the ImageNet-1k pretrained models. More interestingly, these graph measure changes also corresponded to the model generalizability, which decreased as the average path length increased and the clustering coefficient decreased. These findings were consistent with those in Fig. 7.2 and further indicated that the graph measures could reflect not only the model classification performance, but also the model sampling efficiency.

7.5.3 Aggregation Graphs between Layers

In addition to the overall aggregation graph of different ViT models, we also investigated the aggregation graphs of different layers of the model. As shown in Fig. 7.9, the clustering coefficient tends to increase with the number of layers (Fig. 7.9(a)), which was further demonstrated by a visible example (ViT-B) in Fig. 7.9(b). The clustering coefficients reflected the degree of aggregation of the network that low-level features continuously aggregated into high-level features as the model layers became deeper. Our findings are consistent with previous studies [289] from a graph measure perspective.

7.5.4 Impact on Model Design and Optimization of ANNs

The proposed relational graph provides a unified paradigm to represent various ViT models, thus enabling effective comparisons among different ANNs and their associations with BNNs. Given that the brains are already highly optimized BNNs and these structural and functional BNNs possess a variety of interesting and nice graph properties, such as the aforementioned sweet spots, we premise that BNNs' graph characteristics could provide potential guidance for the design and optimization of

ANNs, e.g., through the structural characteristics of the brain and neural architecture search (NAS), and could offer benchmarks for evaluating those optimized ANNs. Also, additional graph models, abstractions, and common frameworks could be explored and summarized from BNNs and then used to inform and guide the design and optimization of next-generation ANNs and their wide applications in the future.

7.5.5 Limitation

7.5.5.1 Limitation on Image Content

The spatial information exchange in ViT is often associated with the content of images. However, it is difficult to integrate specific data information into the graph construction of networks. Here, we made an assumption that the patches of images are independent of each other and the spatial relationship is retained only through positional information. However, some models such as Mixer [272] and ResMLP [276] did not use position embedding and might be sensitive to the order of the input tokens [272]. Therefore, integrating image content into the model structure deserves future exploration.

7.5.5.2 Affine Graph in Transformers

In this paper, we mainly consider the differences of different models Aggregation Graph without focusing on Affine Graph. One of the main reasons is that almost all vision transformers use the same Channels MLP (e.g. two fully connected linear with GeLU) for the communication between channels. Another reason is that there is no suitable solution to the problem of different width dimensions. Unlike visual images which are strongly local, thus allowing us to do sampling operations on images, the interaction between features is often global and it is difficult to unify the size. In the future, there is still a requirement to explore more effective ways to combine these two kinds of graphs.

7.6 Conclusion

In this paper, we proposed a novel unified and biologically-plausible relational graph representation of vision transformers. By decomposing the information transfer graph of the network into the aggregation and affine graphs, our method can be applied to almost all representative visual models. By exploring the differences in graph measures of aggregation graphs in different ViTs, we found that the model performance is closely related to the graph measures, especially when the sample size is small. We also found that the proposed relational graph representation of ViTs has high similarity with real BNNs derived from brain science data. Finally, we also proposed a graph loss to further improve the model performance. Overall, we provided an interpretable and effective way for analyzing ViT models and linking models' relational graphs with BNNs, and offered novel insights on the design of ANNs.

CHAPTER 8

CORE-PERIPHERY PRINCIPLE GUIDED REDESIGN OF SELF-ATTENTION IN TRANSFORMERS

The research work introduced in **Chapter 7**, by post-hoc analysis, have found that the best-performing ANNs surprisingly resemble biological neural networks (BNNs), which indicates that ANNs and BNNs may share some common principles to achieve optimal performance in either machine learning or cognitive/behavior tasks. Inspired by this phenomenon, rather than relying on post-hoc schemes, we proactively instill organizational principles of BNNs to guide the redesign of ANNs. We leverage the Core-Periphery (CP) organization, which is widely found in human brain networks, to guide the information communication mechanism in the self-attention of vision transformer (ViT) and name this novel framework as CP-ViT. In CP-ViT, the attention operation between nodes (image patches) is defined by a sparse graph with a Core-Periphery structure (CP graph), where the core nodes are redesigned and reorganized to play an integrative role and serve as a center for other periphery nodes to exchange information. In addition, a novel patch redistribution strategy enables the core nodes to screen out task-irrelevant patches, allowing them to focus on patches that are most relevant to the task.

8.1 Introduction

Aided by the rapid advancement in hardware and massively available data, deep learning models have witnessed an explosion of various artificial neural networks (ANN) architectures[90, 290, 274], and made breakthroughs in many application fields due to their powerful automatic feature extraction capabilities. It is widely expected the architectures of ANN, as the core of current AI techniques, to be more efficient, reliable, explainable, and transformable, to adapt to various and complex problems in real applications. Essentially, various ANN architectures, represented via different neuron wiring patterns, correspond to different information exchange mechanisms, and therefore, have an inevitable effect on the latent feature representation and the downstream task performance. For example, multilayer perceptron (MLP) directly stacks multiple layers of neurons with paired-wise full connections between adjacent layers, whereas convolutional neural networks (CNN) focus on learning effective convolutional kernels that indicate specific wiring patterns among the neurons within the receptive field. Similarly, recurrent neural networks (RNN) adopt cyclic connections between nodes, allowing output to affect subsequent input to the same nodes[291]. This special neuron wiring pattern of building cycles between nodes also enables RNNs to model and infer temporal dynamic relationships[292] contained in sequential data. More recently, transformer has become another mainstream ANN architecture due to its outstanding self-attention mechanism that allows effective and efficient message exchanges among neurons, and produced promising results in the natural language processing[274, 179] and computer vision domains[5, 275]. In particular, many advancements in transformer architecture design, e.g., vision transformer (ViT)[5], have centered around more effective message exchange mechanisms among spatial tokens by designing different Token Mixers. For instance, the shifted window attention in Swin[275], the token-mixing MLP in Mixer[272], and the pooling in MetaFormer[293], among others, were all designed to improve the self-attention upon the original vanilla ViT[5], and thus enable more effective and efficient message exchanges among spatial patches/tokens. However, despite tremendous advancements in ANN architecture design in MLPs, CNNs, RNNs, and transformers, particularly for better message exchange mechanisms, there has been a fundamental lack of general principles that can inform and guide such ANN architecture design and redesign.

To seek such guiding principles for ANN architecture design, more and more research studies started exploring the “structural signatures” of well-performing ANNs. Hence, the deep learning community has witnessed a paradigm shift from optimal feature design to optimal ANN architecture design. In general, the major strategies for optimal ANN architecture design can be categorized into two basic streams based on how to search in the neural architecture space. The first strategy is to design neural architectures that achieve the best possible performance using given computing resources in an automated way with minimal human intervention. Neural architecture search (NAS)[294, 295, 296] is a major methodology in this category. NAS has a relatively low demand for the researchers’ prior knowledge and experience, making it easier to perform modifications to the neural architecture though it usually comes with a high computational cost. The second category of the strategy is to take the advantage of prior knowledge from specific domains, such as brain science, to guide ANN architecture design. For example, the authors in [297] designed a two-stream model for grounding language learning in vision based on the brain science principle that humans learn language by grounding concepts in perception and action, and encoding “grounded semantics” for cognition. It is worth noting that the above-mentioned two strategies should be viewed as complementary to each other rather than being in conflict, and their combination provides the researchers with an opportunity to explore and design well-performing neural architectures under different principles. For instance, recent studies, via qualitatively post-hoc analysis, have found that the best-performing ANNs surprisingly resemble biological neural networks (BNN)[12], which indicates that ANNs and BNNs may share some common principles to achieve optimal performance in either machine learning or cognition/behavior tasks.

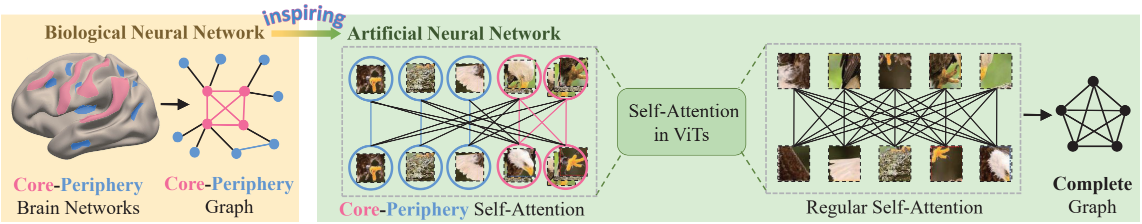

Biological Neural Network
inspiring
Artificial Neural Network
Core-Periphery Brain Networks
Core-Periphery Graph
Core-Periphery Self-Attention
Self-Attention in ViTs
Regular Self-Attention
Complete Graph

Figure 8.1. The Core-Periphery principle in brain networks inspires the design of ANNs. The Core-Periphery structure broadly exists in brain networks, with a dense "core" of nodes (pink) densely interconnected with each other and a sparse "periphery" of nodes (blue) sparsely connected to the core and among each other. Inspired by this principle of BNN, we aim to instill the Core-Periphery structure into the self-attention mechanism and propose a new CP-ViT model.

Inspired by the above-mentioned prior outstanding studies, in this work, we aim to proactively instill the Core-Periphery (CP) organization to guide the redesign of ANNs by using ViT as a working example. It has been widely confirmed that the Core-Periphery organization universally exists in the functional networks of human brains and other mammals, effectively promoting the efficiency of information transmission and communication for integrative processing[298, 299]. The concept of the Core-Periphery brain network is illustrated in Fig. 8.1. By using the Core-Periphery property as a guiding principle, we infused its effective and efficient information communication mechanism into the redesign of ViT. To this end, we quantified the Core-Periphery property of the human brain network, infused the Core-Periphery property into ViT, and proposed a novel CP-ViT architecture. Specifically, we update the complete graph of dense connections in the original vanilla ViT[5] with a sparse graph with Core-Periphery property (CP graph), where the core nodes are redesigned and reorganized to play an integrative role and serve as a center for other periphery nodes to exchange information. Moreover, in our design, a novel learning mechanism is used to endow the core nodes with the power to capture the task-related meaningful and important image patches. We evaluated the proposed CP-ViT on multiple public datasets, including a medical image dataset (INbreast) and natural image datasets (CIFAR-10, CIFAR-100, TinyImageNet). The results indicate that the optimized CP-ViT in sweet spots[12] outperforms other ViTs. We summarize our contributions in three aspects: 1) This work provides novel insights for brain-inspired AI: we can utilize the principles found in BNNs to guide and improve our ANN architecture design; 2) We show that there exist sweet spots of CP graphs that lead to CP-ViTs with significantly improved performance and 3) The core nodes in CP-ViT correspond to task-related meaningful and important image patches, which can significantly enhance the interpretability of the trained deep model.

8.2 Methods

8.2.1 Related Work

Core-periphery Structure The Core-Periphery structure is a fundamental network signature that is composed of two qualitatively distinct components: a dense "core" of nodes strongly interconnected with one another, allowing for integrative information processing to facilitate the rapid transmission of the message, and a sparse "periphery" of nodes sparsely connected to the core and among each other[300]. The Core-Periphery pattern has helped explain a broad range of phenomena in network-related domains, including online amplification[301], cognitive learning processes[298], technological infrastructure organization[302, 303], and critical disease-spreading conduits[304]. All these phenomena suggest that the Core-Periphery pattern may play a critical role to ensure the effectiveness and efficiency of information exchange within the network. In the literature, there are two widely-used approaches for generating graphs with Core-Periphery property (CP graphs): the classic two-block model of Borgatti and Everett (BE algorithm)[305] and the kcores decomposition[300]. The former approach partitions a network into a binary hub-and-spoke layout, while the latter one divides it into a layered hierarchy. In this work, for simplicity, we adopted a two-block model to generate a CP graph which is used to guide the self-attention operations between patches (tokens) in ViT. In this way, the Core-Periphery property is infused into the ViT model.

Methods for Designing More Efficient ViT Architecture ViT and its variants have achieved promising performances in various computer vision tasks, but their gigantic parameter counts, heavy run-time memory usage, and high computational cost become a major burden for the applications. Therefore, there is an urgent need to develop lightweight vision transformers with comparable performance and efficiency. For this purpose, several studies aimed to use network pruning, sparse training, and supernet-based NAS to slim vanilla ViT. **From token level**, Tang et al.[306] designed a patch slimming method to discard useless tokens. Evo-ViT[307] updated the selected informative and uninformative tokens with different computation paths. VTP[308] reduced embedding dimensionality by introducing control coefficients. **From model architecture level**, UP-ViTs[309] pruned the channels in ViTs in a unified manner, including residual connections in all the blocks, multi-head self-attention (MHSA)[274], feedforward neural layers (FFNs), normalization layers, and convolution layers in ViT variants. SViTE[310] dynamically extracted and trained sparse subnetworks instead of training the entire model. To further co-explore data and architecture sparsity, a learnable token selector was used to determine the most vital image patch embeddings in the current input sample. AutoFormer[311] and ViTAS[312] leveraged supernet-based NAS to optimize the ViT architecture. Despite the remarkable improvements achieved by the above methods, both token-sampling and data-driven strategies may highly depend on the data and tasks performed, impeding the vision transformers' generalization capability. A more universal principle (e.g., derived from BNNs) that can guide a more efficient design of ANN's architecture is much desired. In this work, we will leverage a widely existing Core-Periphery property in BNN to develop a more efficient CP-ViT.

8.2.2 Core-Periphery Principle Guided Transformer

The Core-Periphery principle can be applied to ViT and its variants via a unified framework that is illustrated in Fig. 8.2. The framework includes two main parts: Core-Periphery graph generation and Core-Periphery graph guided re-design of the self-attention mechanism.

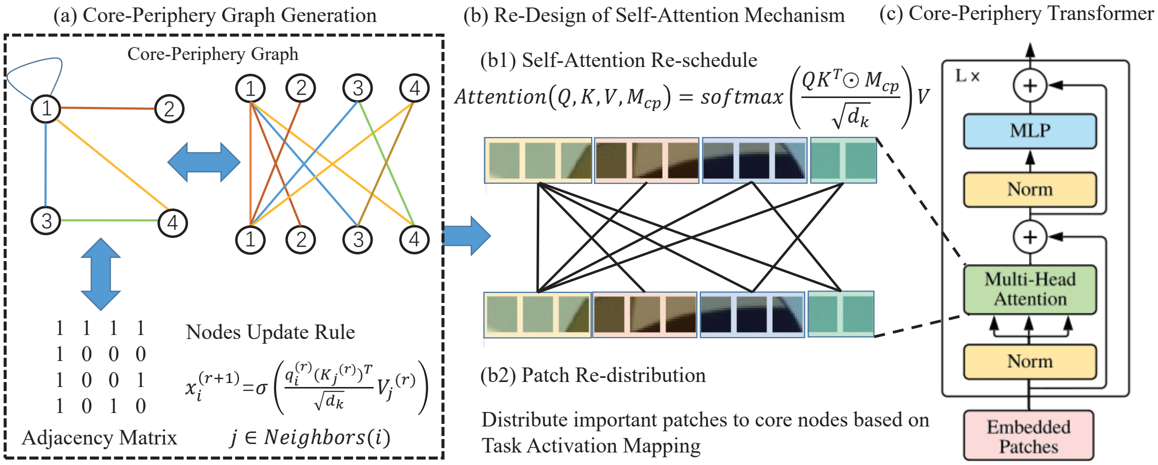

(a) Core-Periphery Graph Generation
Core-Periphery Graph
1
2
3
4
1 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
Adjacency Matrix
Nodes Update Rule
$x_i^{(r+1)} = \sigma\left(\frac{q_i^{(r)}(K_j^{(r)})^T}{\sqrt{d_k}} V_j^{(r)}\right)$
$j \in Neighbors(i)$
(b) Re-Design of Self-Attention Mechanism
(b1) Self-Attention Re-schedule
$Attention(Q,K,V,M_{cp}) = softmax\left(\frac{QK^T \odot M_{cp}}{\sqrt{d_k}}\right)V$
(b2) Patch Re-distribution
Distribute important patches to core nodes based on Task Activation Mapping
(c) Core-Periphery Transformer
L ×
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

Figure 8.2. Core-Periphery Principle Guided Re-design of Self-Attention. The proposed Core-Periphery guided re-design framework for ViTs consists of two major components: the Core-Periphery graph generator and the re-design of the self-attention mechanism. The basic idea is that we mapped the ViT structure to graphs and proposed a new graph representation paradigm to represent the self-attention mechanism. Under this paradigm, the design of the self-attention mechanism can be turned into a task of designing desirable graphs. (a) The CP graph generator was proposed to generate graphs with Core-Periphery property in a wide range of search spaces. (b) The self-attention of the nodes is controlled by the generated CP graph and the patches are re-distributed to different nodes by a novel patch distribution method. (c) The new self-attention mechanism will upgrade the regular self-attention in vanilla ViT. The new ViT architecture is thus named as CP-ViT.

8.2.2.1 Core-Periphery Graph Generation

The self-attention of our proposed CP-ViT is controlled by Core-Periphery graphs (CP graphs). We proposed a CP graph generator to generate a wide spectrum of CP graphs in the graph space defined by the number of total nodes and the core nodes. Although several graph generators have been proposed in previous works, they were not designed for generating CP graphs. For example, Erdos-Renyi (ER) generator samples graphs with given node and edge numbers uniformly and randomly[313]; Watts-Strogatz (WS) generator generates graphs with small-world properties[314], and the complete graphs generator generates graphs where nodes are pair-wise densely connected with each other[315].

To generate graphs with CP property, we proposed a novel CP graph generator that is parameterized by a total node number n, a core node number m, and three wiring thresholds p_{cc}, p_{cp}, p_{pp} which are the wiring probabilities between the core-core nodes, core-periphery nodes, and periphery-periphery nodes, respectively. Based on these measures, the CP graph generation process is as follows: we first defined the core nodes number m and the periphery nodes number $n - m$; Then, for each of the core-core node pairs, we used a random seed sampled from the continuous uniform distribution in $[0, 1]$ to generate a wiring probability p_{rs}. If the wiring probability is greater than the threshold p_{cc}, the two core nodes are connected. This wiring process is formulated as:

$$A\left(i,j\right)=\begin{cases}1 & \text{if } p_{rs}\geq p_{cc}\\ 0 & \text{if } p_{rs}<p_{cc}\end{cases} \tag{8.1}$$

where A is the adjacency matrix of the generated graph, 1 means that there exists an edge between the nodes i and j, 0 means there is no edge between the nodes. The same procedure was applied to core-periphery and periphery-periphery node pairs with the corresponding thresholds p_{cp} and p_{pp}, respectively. In this way, by using different combinations of n, m, and wiring thresholds, we can generate a large number of candidate graphs in the graph space; finally, all the generated graphs were examined by the CP detection algorithm (BE algorithm)[305] and the graphs with CP property will be used in the further steps to guide the self-attention operation.

8.2.2.2 Core-Periphery Guided Self-Attention

To instill the CP principle into the self-attention mechanism in ViT, we redesigned the self-attention operations according to the generated CP graphs: the patches are replaced by the nodes, and the new self-attention relations are replaced by the edges in the CP graph. Thus, the self-attention in the vanilla ViT can be represented as a complete graph, and similarly, the CP principle can be effectively and conveniently infused into the ViT architecture by upgrading the complete graph with the generated CP graphs. CP graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with nodes set $\mathcal{V}$ and edges set $\mathcal{E}$. The redesign of self-attention is formulated as:

$$x_i^{(r+1)} = \sigma^{(r)}(\{(\frac{q_i^{(r)}(K_j^{(r)})^T}{\sqrt{d_k}})V_j^{(r)}, \forall j \in N(i)\})(8.2)$$

where $\sigma(\cdot)$ is the activation function, which is usually the softmax function in ViTs, $q_i^{(r)}$ is the query of patches in the i-th node in $\mathcal{G}$, $N(i) = \{i \| i \vee (i,j) \in \mathcal{E}\}$ are the neighborhood nodes of node i, d_k is the dimension of queries and keys, and $K_j^{(r)}$ and $V_j^{(r)}$ are the key and value of patches in node j.

In vanilla ViT, one input image is divided into 196 patches, and each patch resolution is 16 by 16. In CP-ViT, each node corresponds to a single patch or multiple patches. We proposed the following patch assignment pipeline to map the original patches to the nodes: for a CP graph with n nodes, each node will be assigned to either $\lfloor 196/n \rfloor + 1$ or $\lfloor 196/n \rfloor$ patches. For example, if we use a CP graph with 5 nodes, the 5 nodes will have 40, 39, 39, 39, and 39 patches, respectively; and if we use a CP graph with 196 nodes, each node will correspond to a single patch. Note that the patches are randomly assigned to the nodes at the beginning of the training process, and then they will be re-distributed iteratively after each training epoch based on a novel patch distribution method that will be elaborated in the next section. Based on the above discussion, the CP graph-guided self-attention conducted at the node level can be formulated as:

$$Attention(Q, K, V, M_{cp}) = softmax(\frac{QK^T \odot M_{cp}}{\sqrt{d_k}}V) \quad (8.3)$$

where the queries, keys, and values of all the patches are packed into the matrices Q, K, and V, respectively. M_{cp} is the mask matrix derived from the adjacency matrix A of the CP graph, and $\odot$ is the dot product. The size of the mask matrix M_{cp} is 197×197 (196 patches plus 1 classification token), and it is a symmetric matrix. The derivation process of M_{cp} is as follows: for a CP graph with 5 nodes, the 5 nodes have 40, 39, 39, 39, and 39 patches, respectively. If the element $(1, 2)$ in the corresponding adjacency A is 1, which means the node #1 is connecting to the node #2, and as a result, the 40 patches corresponding to the node #1 are connecting to the 39 patches associated with the node #2. Therefore, the elements at $(1 : 40, 40 : 79)$ and $(40 : 79, 1 : 40)$ in the mask matrix M_{cp} will be 1, where the $(40 : 79, 1 : 40)$ means the elements from the 40th row to 79th row, and from the 1st column to the 40th column. The elements in the last row and column of M_{cp} are 1 because the classification token is connected to all the nodes, including both core and periphery nodes. Similar to the multi-head attention in transformers[274], our proposed CP multi-head attention is formulated as:

$$\begin{aligned} MultiHead(Q, K, V, M_{cp}) &= Concat(head_1, ..., head_h)W^o \\ where\ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V, M_{cp}) \end{aligned} \quad (8.4)$$

where the parameter matrices W_i^Q, W_i^K, W_i^V and W^O are the projections. Multi-head attention helps the model to jointly aggregate information from different representation subspaces at various positions. In this work, we apply the CP principle to each representation subspace.

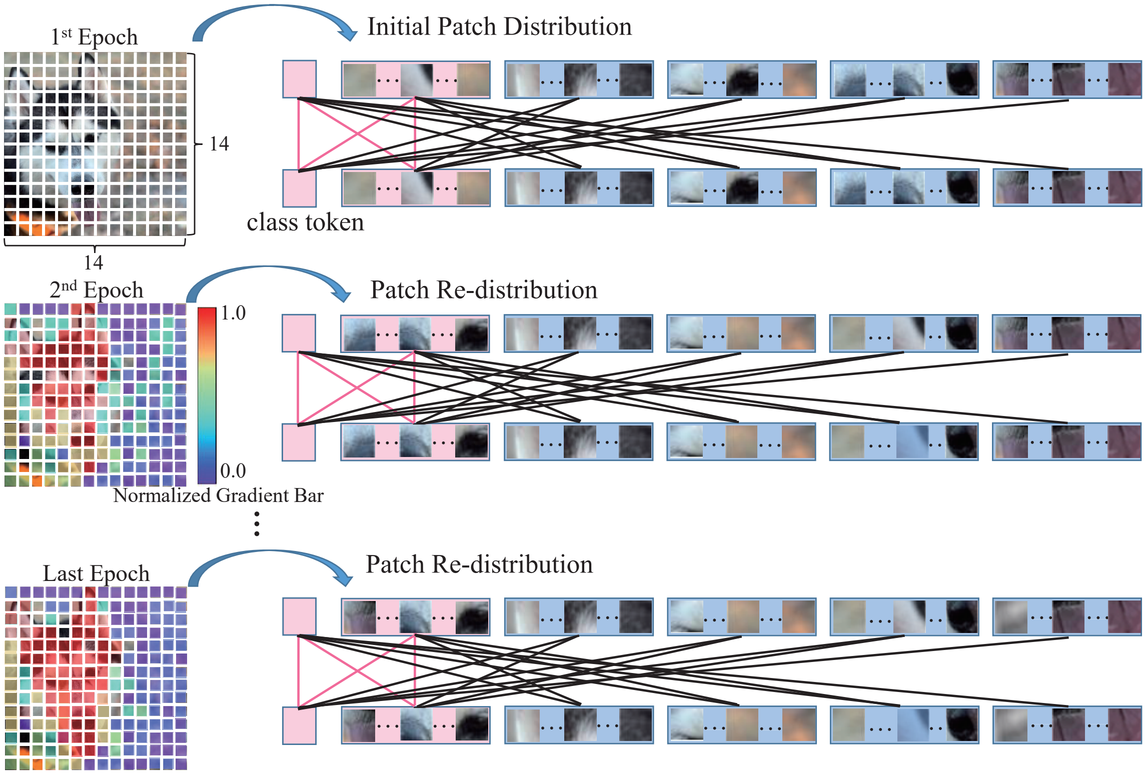


Figure 8.3. Illustration of Patch Redistribution Process. The pink nodes are the core nodes, while the blue nodes are the periphery nodes. The initial patch distribution at the first epoch is the same as the vanilla ViTs. After each iteration during the training process, the gradients of patches discriminate from each other due to different contributions to the classification. The red the image patches are, the high gradient they are. Thus, the core patches that contribute most to the classification task are re-distributed to core nodes.

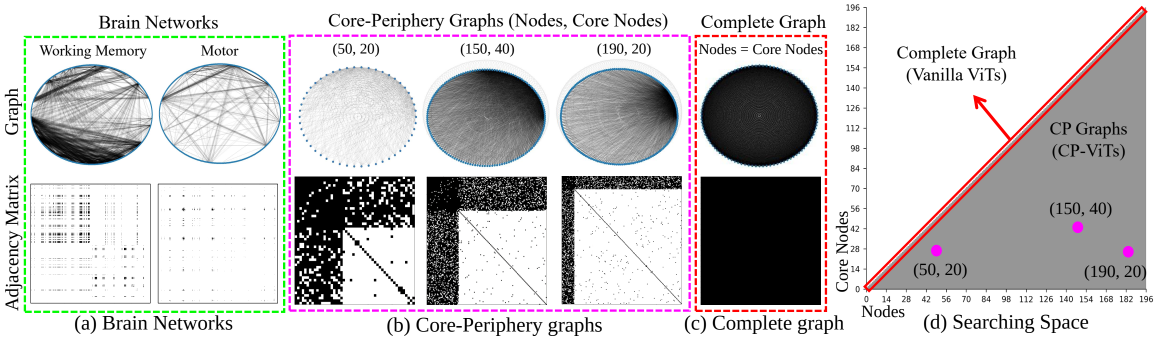
8.2.2.3 Patch Redistribution

The CP structure inclines to make the communication and message exchange at core nodes more intensive while less frequent among periphery nodes. This is based on the fact that the core nodes usually process the most important information in many biological networks[298]. To this end, we need to evaluate the importance of the patches and select the most important ones to assign to the core nodes, which is defined as task-related activation feature mapping. For a specific task of CP-ViT, in order to identify the important patches, we computed the gradients of the output y (before the activation function) with respect to patch features (after patch embedding) P^k, i.e. $\frac{\partial y}{\partial P^k}$. These gradients flowing back to the patch features are global-average-pooling over the feature dimensions to obtain the patch importance weights. The important weights are:

$$\alpha_k = \frac{1}{Z} \sum_{i=1}^{Z} \frac{\partial y}{\partial P_i^k} (8.5)$$

where Z is the dimension of the patch embedding features. After we have the weights of all the patches, the top K patches that have the highest weights are selected and re-distributed to the core nodes. Note that the patch distribution process is not random but distributed based on the nodes' degree in a in a descending manner: the patches with higher importance weights are distributed to the core nodes with a higher degree. The corresponding patch redistribution process is illustrated in Fig. 8.3. As shown in Fig. 8.3, the image patches were randomly distributed at the first epoch but as the training process proceeded, patches with high gradients are identified as important patches and gradually redistributed to the core nodes. After certain iteration epochs, those patches that contribute the most to the classification result will be distributed to the core nodes.

8.3 Results



(a) Brain Networks (b) Core-Periphery graphs (c) Complete graph (d) Searching Space

Figure 8.4. (a) Two types of representative brain networks in motor and working memory tasks. (b) Three examples of CP graphs. (c) Complete graph. The first row in (a), (b), and (c) shows their wiring patterns, while the second row shows their corresponding adjacency matrices. Black color in adjacency matrices means connections between nodes, while white represents no edge. (d) Graph search space defined by the total nodes number and the core nodes number. The complete graphs are located at the diagonal highlighted by a red box and the CP graphs are located at the remaining parts.

8.3.1 Exploring Core-Periphery Graphs

Core-Periphery property in brain networks. We quantitatively measured the Core-Periphery property of brain networks. Working memory network (BN-WM) and motor network (BN-M) are two typical functional networks that are widely existed in the human brain. In this work, we used task fMRI data of these two tasks in the Human Connectome Project[316] to generate functional brain networks. Using voxels as nodes and the correlations between fMRI signals associated with each voxel as edges, we built two population-level functional networks and showed their connection patterns as well as the adjacency matrices in Fig. 8.4(a). To measure the Core-Periphery property of the two functional brain networks, we adopted independent probability[317] as the measurement. Independent probability is defined as the probability that there is an edge between any pairs of nodes in a given matrix. Thus, the independent probabilities of the core-core connections, core-periphery connections, and periphery-periphery connections can be represented as I_{cc}, I_{cp} and I_{pp}, respectively. If the given matrix or graph is organized in a Core-Periphery manner[318][319], the corresponding independent probabilities will have the following relations: $I_{cc} > I_{cp} > I_{pp}$. According to previous studies[320], the convex gyri and concave sulci areas, which are two basic anatomical structures of the cerebral cortex, play different functional roles: gyri are functional hubs for global information exchange while sulci are responsible for local information processing. Therefore, we divided the nodes (voxels) into two categories, gyri-nodes (nodes in gyri regions) and sulci-nodes (nodes in sulci regions), and examined if brain networks have CP structure: gyri-nodes act as core nodes and sulci-nodes act as periphery nodes. The core-periphery measures of brain networks are shown in the last two columns in Table 8.1. R_{cc}, R_{pp} and R_{cp} represent the normalized independent probabilities of core-core, core-periphery, and periphery-periphery connections. The independent probabilities and normalized independent probabilities are formulated as:

$$\begin{aligned} I_{cc} = \frac{1_{A_{cc}}}{\|A_{cc}\|_1}, I_{cp} = \frac{1_{A_{cp}}}{\|A_{cp}\|_1}, I_{pp} = \frac{1_{A_{pp}}}{\|A_{pp}\|_1}, \\ R_{cc} = I_{cc}/(I_{cc} + I_{cp} + I_{pp}), \\ R_{cp} = I_{cp}/(I_{cc} + I_{cp} + I_{pp}), \\ R_{pp} = I_{pp}/(I_{cc} + I_{cp} + I_{pp}). \end{aligned} \tag{8.6}$$

Core-Periphery structure in artificial neural networks. We introduced the Core-Periphery organization into ANNs by CP graphs. There are two key factors that can affect the CP graph generation process. The first is the number of nodes, including the number of total nodes and the core nodes, which defines the search space. In this work, we set the maximum number of total nodes as 196, i.e., the number of patches for the vision transformer, then the number of core nodes can be any number between 0 and 196. Thus, the search space will include $\sum_{i=1}^{196}\sum_{j}^{0<j<=i}(i+j) = 19208$ types of CP graphs, where i and j represent the number of total nodes and the core nodes. The second is the wiring patterns of CP graphs: in this work, we used p_{cc}, p_{cp}, and p_{pp} to represent the wiring probabilities between core-core nodes, core-periphery nodes, and periphery-periphery nodes, respectively. Fig. 8.4 (b) and (c) present the wiring patterns and adjacency matrices of three examples of CP graphs and the complete graph. As shown in Fig. 8.4(b) and (c), CP graphs are densely connected for core nodes and sparsely connected for periphery nodes. The overall connection patterns of CP graphs are more sparse than the complete graph. The search space of CP graphs was shown in Fig. 8.4(d) where the complete graphs located at the diagonal were highlighted by a red box and three types of CP graphs corresponding to Fig. 8.4(b) were highlighted by pink circles. For each type of CP graph, we generated 5 samples with different wiring patterns and obtained 19208 * 5 CP graphs in total. Since the number of the generated CP graphs is huge (19208 * 5 in total), we sampled 190 types of CP graphs out of the total 19208 and finally obtained 190*5 candidates. For example, for a CP graph with 50 nodes, the number of core nodes is set to be [10, 20, 30, 40]. As a result, four different CP graphs, including [50, 10], [50, 20], [50, 30], and [50, 40], are obtained. For each of these four types of CP graphs, we generate 5 samples for further experiments.

Similar to brain networks, we also used the normalized independent probability to measure the Core-Periphery property for the generated CP graphs. We calculated the normalized averaged independent probability over 190*5 CP graphs and showed the results in the first column of Table 8.1. From the table we can see that $R_{cc} > R_{cp} > R_{pp}$, which suggests that our generated CP graphs, as expected, display prominent Core-Periphery properties, while the graphs generated by the classic graph generators,

Table 8.1. Evaluation of the Core-Periphery property in CP graphs, graphs generated by other graph generators, and brain networks

IP	CP Graphs	CE. Graphs	WS Graphs	ER Graphs	BN-M	BN-WM
R_{cc}	$.59 \pm .06$	$.33 \pm .00$	$.40 \pm .27$	$.36 \pm .23$	$.55 \pm .11$	$.61 \pm .09$
R_{cp}	$.35 \pm .13$	$.33 \pm .00$	$.40 \pm .28$	$.36 \pm .24$	$.34 \pm .07$	$.26 \pm .10$
R_{pp}	$.07 \pm .06$	$.33 \pm .00$	$.20 \pm .28$	$.28 \pm .22$	$.15 \pm .05$	$.14 \pm .06$

such as (1) Complete graph (CE.) generator; (2) Watts-Strogatz (WS) generator; and (3) Erdos-Renyi (ER) generator don't have the Core-Periphery property.

Table 8.2. Summary of datasets

Dataset	Training	Validation	Class	Original Res.	Resized Res.
INbreast	6000	100	3	1024 * 1024 * 3	224 * 224 * 3
CIFAR-10	50000	10000	10	32 * 32 * 3	224 * 224 * 3
CIFAR-100	50000	10000	100	32 * 32 * 3	224 * 224 * 3
TinyImageNet	100K	10000	200	64 * 64 * 3	224 * 224 * 3

8.3.2 Sweet Spots for CP-ViTs

In this section, we evaluated the performance of the proposed CP-ViT. The CP-ViT was implemented based on the ViT-S/16 architecture[321] and evaluated on 4 different types of public datasets, the medical image dataset INbreast[322], the natural image dataset CIFAR-10[280], CIFAR-100[280] and TinyImageNet[323]. The summary of the datasets we used in this work is presented in Table 8.2. The parameters of CP-ViT were initialized and fine-tuned from ViT-S/16 trained on ImageNet[290]. We trained the CP-ViT for 100 epochs with batch size 64 for INBreast and 256 for CIFAR-10, CIFAR-100 and TinyImageNet, and used AdamW optimizer and cosine learning rate schedule[324] with an initial learning rate of 0.0001 and minimum of 1e−6. All the experiments were conducted using NVIDIA Tesla V100 GPU.

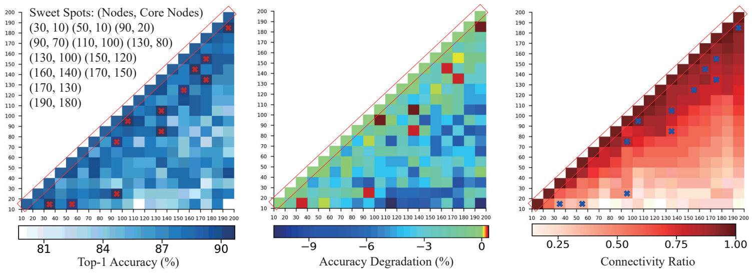
We explored the performance of different types of CP graphs in the search space (Fig. 8.4(a)) in terms of top 1 accuracy and connection ratio. The connection ratio (CR) quantitatively measures the computational costs of different self-attention operations, which is defined by (8.7):

$$CR = \frac{1_{M_{cp}}}{\|M_{cp}\|_1}(8.7)$$

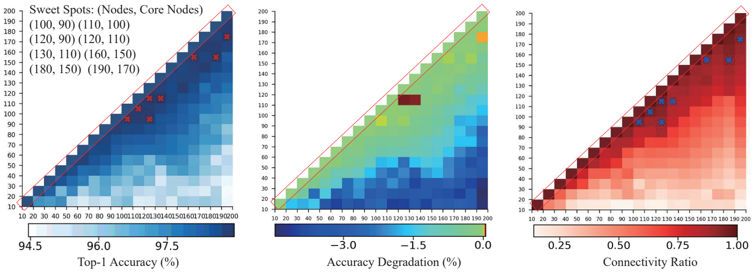
where $1_{M_{cp}}$ represents the number of 1s in the mask matrix of cp graphs - M_{cp} which is derived from the adjacency matrix of the CP graph, and $\|\bullet\|_1$ is the number of elements in the mask matrix. In general, CR represents the ratio of actual self-attention operations to the potential maximum self-attention operations. Given a graph, the potential maximum self-attention operation is fixed. Less actual self-attention operation means less computational cost and hence it has a smaller CR value.

For each specific combination of different numbers of nodes/core nodes in the search space, we trained the CP-ViT with 5 different CP graph samples and reported the average result in Fig. 8.5. The four results in Fig. 8.5(a-d) correspond to four different datasets. For the results on each dataset, we display three subfigures: the top 1 accuracy (left), the accuracy degradation (middle), and the connection ratio (right). We highlighted the sweet spots, which are corresponding to the CP graphs that lead to improved performance[12], with red crosses in Fig. 8.5. In the top-1 accuracy of Fig. 8.5, deeper color means better performance. The accuracy degradation subfigures show the accuracy variation compared to fully connected self-attention ViTs. Our CP-ViTs gain a positive boost in sweep spots as it has higher accuracy than vanilla ViTs. At the same time, our CP-ViTs maintain competitive top-1 accuracy in most

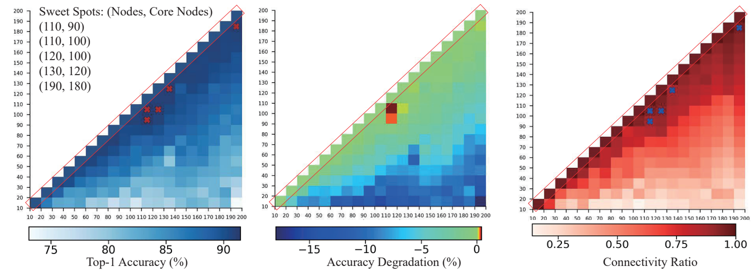
(a) CP-ViT on INbreast



(b) CP-ViT on CIFAR-10



(c) CP-ViT on CIFAR-100



(d) CP-ViT on TinyImageNet

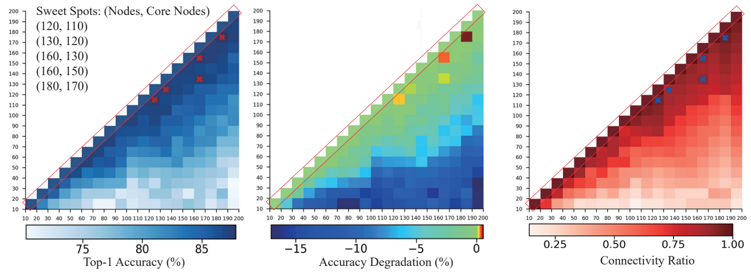


Figure 8.5. Performance of CP-ViT measured using INbreast, CIFAR-10, CIFAR-100 and TinyImageNet datasets. Sub-figures on the left column under each datasets show the top 1 classification accuracy of the CP-ViTs and vanilla ViTs in the search space. A deeper color means higher top 1 accuracy. Sweet spots are marked by red crosses, in which CP-ViTs achieve better performance than vanilla ViT. Sub-figures on the middle column are the accuracy degradation of the CP-ViTs compared to vanilla ViTs. Sub-figures on the right column are the self-attention connection ratio of the CP-ViTs and vanilla ViT. Lighter color means a lower connection ratio. Sweet spots are marked by the blue crosses.

Table 8.3. Comparison between the proposed CP-ViT in sweet spots with finetuned vanilla ViT-S [5]. * means vanilla ViT-S finetuned by ourselves.

Dataset	Model	CP Graph	CR (%)	R_{cc},R_{cp},R_{pp}	Top1 Acc.(%)
INbreast	ViT-S(*)	(N, N)	100.00	0.33, 0.33, 0.33	89.91
	CP-ViT	(30, 10)	32.36	0.58, 0.33, 0.09	90.58
	CP-ViT	(50, 10)	**29.20**	0.53, 0.34, 0.12	90.01
	CP-ViT	(90, 20)	43.82	0.52, 0.36, 0.12	90.58
	CP-ViT	(90, 70)	84.50	0.54, 0.40, 0.06	90.01
	CP-ViT	(100, 90)	92.80	0.49, 0.39, 0.11	**90.69**
	CP-ViT	(130, 80)	31.34	0.58, 0.34, 0.07	90.58
	CP-ViT	(130, 100)	82.94	0.57, 0.36, 0.07	**90.69**
	CP-ViT	(150, 120)	84.18	0.57, 0.41, 0.02	90.01
	CP-ViT	(160, 140)	87.77	0.55, 0.41, 0.03	90.58
	CP-ViT	(170, 130)	80.79	0.57, 0.41, 0.02	90.58
	CP-ViT	(170, 150)	87.65	0.56, 0.41, 0.03	90.12
	CP-ViT	(190, 180)	84.89	0.52, 0.42, 0.05	90.69
CIFAR-10	ViT-S(*)	(N, N)	100.00	0.33, 0.33, 0.33	98.50
	CP-ViT	(100, 90)	92.80	0.49, 0.39, 0.11	98.91
	CP-ViT	(110, 100)	94.49	0.53, 0.42, 0.05	98.91
	CP-ViT	(120, 90)	89.73	0.51, 0.41, 0.08	98.91
	CP-ViT	(120, 110)	94.70	0.49, 0.38, 0.12	98.97
	CP-ViT	(130, 110)	**87.32**	0.56, 0.40, 0.03	**98.97**
	CP-ViT	(160, 150)	90.47	0.54, 0.39, 0.06	98.91
	CP-ViT	(180, 150)	91.79	0.50, 0.42, 0.07	98.91
	CP-ViT	(190, 170)	92.59	0.53, 0.43, 0.03	98.94
CIFAR-100	ViT-S(*)	(N, N)	100.00	0.33, 0.33, 0.33	91.10
	CP-ViT	(110, 90)	88.96	0.59, 0.37, 0.04	91.32
	CP-ViT	(110, 100)	94.49	0.53, 0.42, 0.05	**91.45**
	CP-ViT	(120, 100)	92.40	0.50, 0.41, 0.09	91.15
	CP-ViT	(130, 120)	**87.50**	0.58, 0.32, 0.09	91.11
	CP-ViT	(190, 180)	94.89	0.52, 0.42, 0.05	91.12
TinyImageNet	ViT-S(*)	(N, N)	100.00	0.33, 0.33, 0.33	87.36
	CP-ViT	(120, 110)	94.71	0.49, 0.39, 0.12	87.51
	CP-ViT	(130, 120)	**87.50**	0.58, 0.33, 0.09	87.37
	CP-ViT	(160, 130)	90.02	0.54, 0.44, 0.02	87.40
	CP-ViT	(160, 150)	90.47	0.54, 0.40, 0.06	87.63
	CP-ViT	(180, 170)	95.84	0.50, 0.43, 0.07	**87.84**

search space areas, as shown in the middle subfigures. The performance of CP-ViTs varies in the search space. This result indicates that different self-attention (wiring) patterns may have great influences on the performances of ViTs. Compared to vanilla ViTs with a fully-connected self-attention pattern, the proposed CP-ViT provides the potential for the model to only search for optimal self-attention patterns. The CRs of all the ViTs including vanilla ViTs and CP-ViTs were shown on the right. The CRs of the sweet spots were marked with a blue cross. Besides the improvement in classification accuracy (0.78% for INbreast, 0.47% for CIFAR-10, 0.35% for CIFAR-100, 0.48% for TinyImageNet), the proposed CP-ViT also leads to a great reduction in connection ratio due to less self-attention operations (-70.80% connections for INbreast, -12.68% connections for CIFAR-10, -12.50% connections for CIFAR-100, -12.50% connections for TinyImageNet). The model setting, top 1 accuracy, and CRs of different ViTs were reported in Table 8.3. For all the four datasets, our CP-ViT not only shows improved classification performance but also reduces connection ratio compared to vanilla ViTs. Interestingly, our results demonstrate that the "sweet spots" are corresponding to the wiring patterns (graphs) with CP structures, instead of fully connected self-attention.

We also compared the proposed CP-ViT with the state-of-the-art methods in Table 8.4, including various convolutional networks and transformer architectures. Note that we applied the core-periphery principle to guide the design on small ViT, therefore, the counterparts we compared to in this work are also small-scale transformers and their variants. "– –" means there is no available reports or not applicable. As presented in the table, our method outperforms the CNNs, and a series of variants of transformers on these datasets, suggesting the superiority of the proposed CP-ViTs over the existing methods.

Table 8.4. Comparisons with state-of-the-art transformers and other architectures.

Model	CIFAR-10	CIFAR-100	TinyImageNet	INbreast
ResNet-18[90]	95.55	76.64	67.33	84.34
ResNet-18+Gaze[325]	−−	−−	−−	86.74
ViT-S-SAM[321]	98.20	87.60	87.50	90.20
ViT-S[321]	97.60	85.70	87.40	89.91
DeiT-S[326]	97.50	90.30	86.90	89.90
Mixer-S-SAM[321]	96.10	82.40	85.60	87.60
T2T-ViT-12[327]	98.53	89.63	86.20	88.40
AutoFormer-S[311]	98.50	90.60	87.60	90.10
CP-ViT-S(ours)	98.97	91.45	87.84	90.69

8.3.3 Visualization of Important Patches

Another advantage of CP-ViT is that it can potentially improve the interpretability of the deep-learning models via semi-intervention when linking the explainable concepts contained in the data to the instilled CP structures (section 3.2.3). In our CP-ViT the core nodes are expected to be associated with the important image patches relating to the classification tasks. To evaluate this, we show the patches that were redistributed to the core nodes when the model was well-trained in Fig. 8.6. For INBreast, we randomly selected the images of three subjects in each class and displayed the original images, the images overlaid with important patches, and the images overlaid with the expert's eye gazes in three columns. As shown in the Fig. 8.6, the patches of the core nodes are well co-localized with the locations that were identified as diagnostic biomarkers of the disease in literature publications[328]. We also show the medical physicians' eye gaze maps on these images, given that the eye gaze acquired by eye-tracking equipment is considered the ground truth for identifying important areas in the image. The important patches identified by our CP-ViT highly overlap with the eye gaze maps, demonstrating the correspondence between the core nodes and the task-related concepts, i.e., the

important image patches. For natural image datasets, we also visualized the patches assigned to the core nodes under the black dotted line in Fig. 8.6. It is clear that the objects in the patches of core nodes are semantically related to the class labels.

8.3.4 Fast Search for Sweet Spots

Our proposed CP-ViT aims to achieve better performance more efficiently, by directly updating the initial dense wiring patterns with sparse CP graphs which are widely existing in BNN. Previous studies suggest that in ANN there exist sweet spots that correspond to some specific wiring patterns leading to significantly improved performance[12]. Therefore, it is interesting to investigate the relationship between sweet spots (the ANN structures with better performance) and the introduced CP structure. We conducted intensive experiments to illustrate how the accuracy changes under the CP measurements (in terms of normalized independent probability) and the results are summarized in Fig. 8.7. We found the normalized independent probabilities between core nodes - R_{cc}, core and periphery nodes - R_{cp} and periphery nodes - R_{pp} fall in different range: $[0.45, 0.70]$ for R_{cc}, $[0.25, 0.45]$ for R_{cp}, and $[0.00, 0.15]$ for R_{cc}. Both R_{cc} and R_{cp} display obvious and consistent patterns in terms of the relationship between ANN performance (accuracy) and CP properties: there exists a certain range of CP structures with which the corresponding wiring patterns of ANN can achieve better performance. For example, when the normalized independent probabilities between core and periphery nodes (R_{cp}) fall within the range of $[0.36, 0.42]$, our CP-ViT inclines to have the best accuracy on all four datasets. On the contrary, the normalized independent probabilities between periphery nodes (R_{pp}) show relatively less influence on the overall performance. These results suggest that

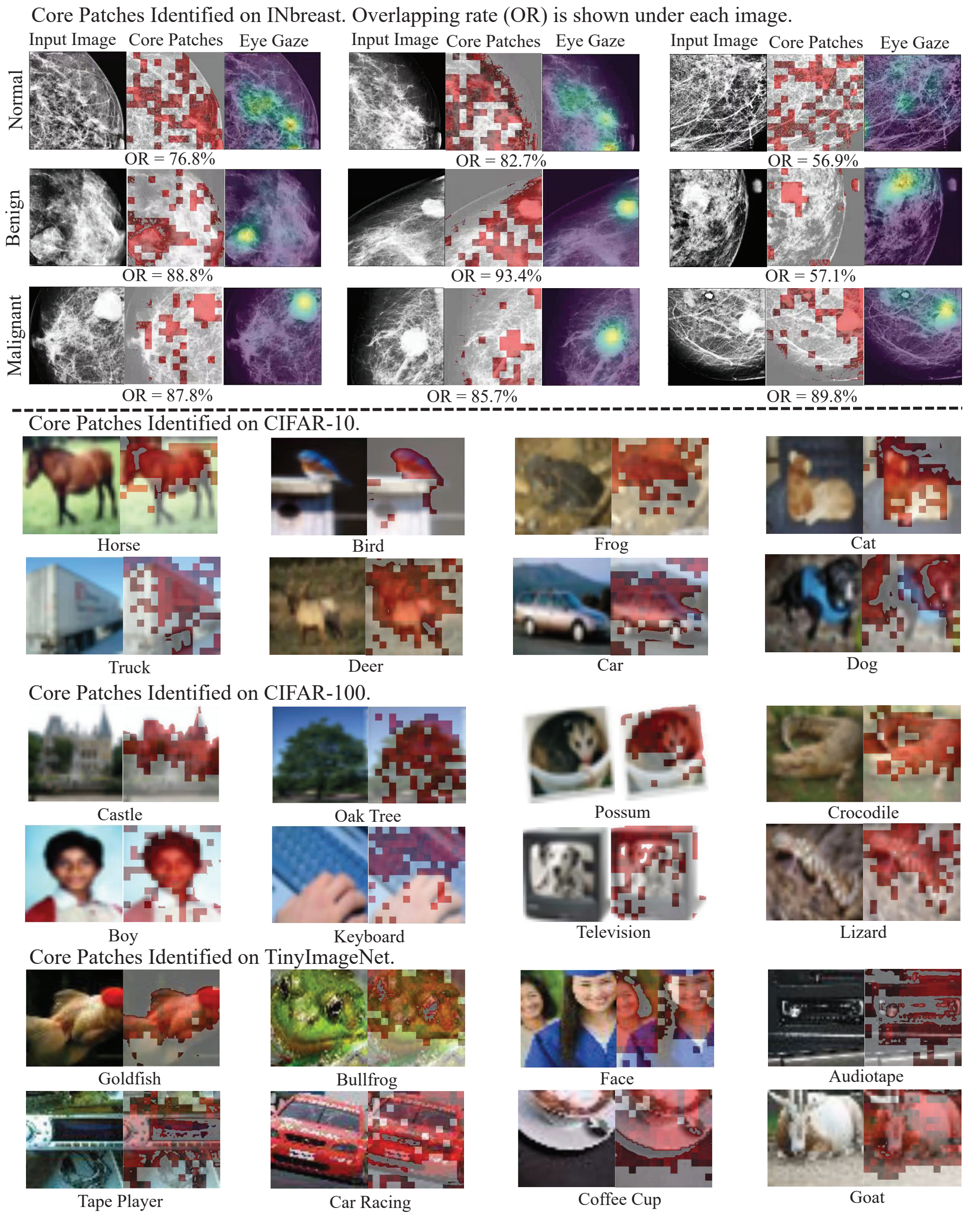
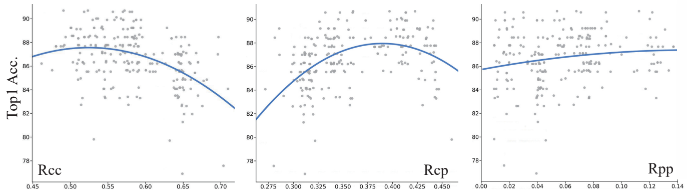
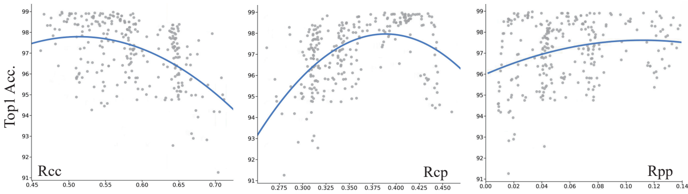


Figure 8.6. Visualization of important image patches that were distributed to the core nodes. For the INbreast dataset (the first block), images of three randomly selected subjects for each class were shown. For each subject, there are three images displayed in three columns. The left column is the original image, the middle column shows the important patches marked by red, and the right column is the eye gaze of medical physicians on the image. For the natural image datasets (the second block, CIFAR-10, CIFAR-100 and TinyImageNet), the important patches identified in eight randomly selected classes were displayed. The left column is the original image, and the right column shows the identified core patches marked in red.

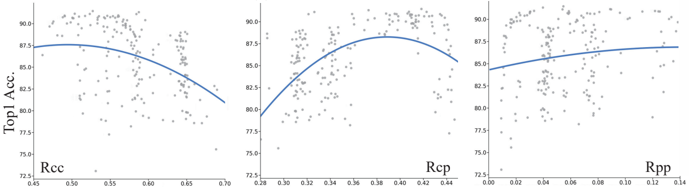
(a) CP-ViT on INbreast



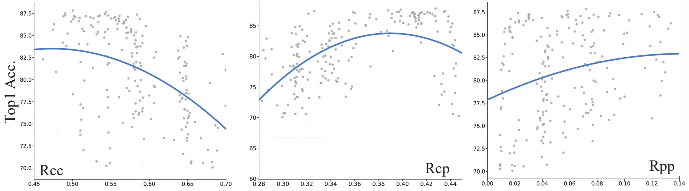
(b) CP-ViT on CIFAR-10



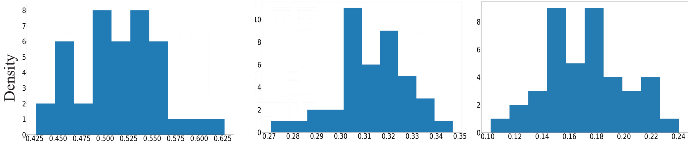
(c) CP-ViT on CIFAR-100



(d) CP-ViT on TinyImageNet



(e) Brain Networks (Motor)



(f) Brain Networks (Working Memory)

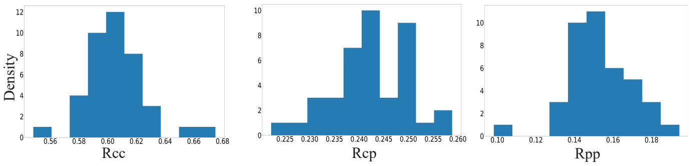


Figure 8.7. Visualization of Core-periphery measures versus the classification performance. The regression results of the normalized independent probability versus the classification accuracy for experiments on each dataset are presented in (a), (b), (c), and (d). The core-periphery measures for brain networks of motor and working memory are shown in (e) and (f).

the wiring patterns between core nodes and periphery nodes have more influence on the overall ANN performance than the wiring patterns between periphery nodes. For comparison, we also calculated the range of group-wise normalized independent probabilities in human functional brain networks when performing two different tasks - motor and working memory tasks. The results are shown in Fig. 8.7 (e-f). Interestingly, the distribution of R_{cc}, R_{cp} and R_{cp} shows obvious overlaps among different functional brain networks though the major range of CP metrics is different from ANN (our CP-ViT). In general, our CP-ViT can leverage the CP structure to learn the optimal combinations of total nodes and core nodes, and to quickly find the sweet spots in a more efficient way.

8.4 Conclusion

In this work, we proactively instilled an organizational principle of BNN, that is, Core-Periphery property, to guide the design of ANN of ViT. For this, we provide a unified framework to introduce the core-periphery principle to guide the design of self-attention, the most prominent mechanism in transformers. Our extensive experiments suggest that there exist sweet spots of CP graphs that lead to CP-ViTs with significantly improved predictive performance. In general, our work advances the state of the art in three ways: 1) this work provides novel insights for brain-inspired AI by applying organizational principles of BNNs to ANN design; 2) the optimized CP-ViT can significantly improve its predictive performance while have the potential to reduce the unnecessary computational cost; and 3) the core nodes in CP-ViT are associated with task-related meaningful image patches, which can significantly enhance the interpretability of the trained deep model.

CHAPTER 9

DISCUSSION AND CONCLUSION

9.1 Conclusion

In general, this thesis aims at bridging the two important domains: brain science and AI, to foster their mutual enhancement. There are two directions: 1) AI in brain science and 2) brain inspired AI.

For AI in brain science, we explore three pivotal problems:

1) Developing graph-based deep generative model to infer individual brain structure-function relationship. Brain's structure-function behaves under a distributed and heterogeneous pattern: at network level, rich functionalities arise from non-overlapping sets of anatomical connections which means functional networks do not necessarily correspond to the underlying structural substrate with a simple node-to-node mapping. Moreover, this complex relation can be interwoven with widely existed individual variations, making it a highly challenging research topic. To solve this problem, we introduced multi-GCN architecture into generative adversarial network (MGCN-GAN) and proposed a novel graph-based GAN model to capture complex individual-specific structure-function relationship. This work is the earliest one in using ANNs to infer brain networks at individual level, which won the Young Scientist Award at the MICCAI 2020.

2) Leveraging multi-modal fusion and embedding learning techniques to advance the study of brain diseases. We proposed a novel graph-based deep neural network – Deep Connectome, to simultaneously model individual structural and functional networks for brain disease analysis. The major advantage of the Deep

Connectome is its capability to integrate multiple types of network connectome and characterize their deep relationship as an “individual connectome signature”. We also developed a novel structure learning method to model the continuum of AD progression as a tree-based trajectory. With the learned tree-based representation of AD progression, we can accurately predict a new individual patient’s clinical status by projecting its brain networks back to this tree structure and examining the projected location within the entire AD spectrum. The two research works are the pioneers in brain disease studies.

3) Exploreing how to unlock the potential benefits of LLMs in brain science and healthcare domains. In our work, we take advantages of the cutting-edge techniques in LLMs domain and solved a longstanding correspondence problem in neuroscience. We also conducted extensive experiments to explore the important problem: whether model designs should be generic or domain specific, which provides new insights for the future development of LLMs.

For brain-inspired-AI, we conducted two-stage study:

1) We conducted post hoc study and built the connection between ANNs and BNNs. Our experimental results suggest that models similar with the more advanced biological brain networks perform better. Therefore, the most advanced human brain is the best reference for designing the optimal architecture of ANNs. This work provides solid experimental support for brian-inspired-AI.

2) Based on the findings in 1), we further incorporated organizational principles of BNNs to guide the design of ANNs. Specifically, we leveraged the widely existed Core-Periphery (CP) organization in BNNs to guide the information communication mechanism of the self-attention in vision transformer (ViT) and propose a novel CP-ViT, which outperforms other state-of-the-art ANNs.

9.2 Future Work

The effective integration of multiple modalities of data is a crucial feature of advanced intelligent systems. In addition, complex real-world problems often involve information from various modalities, especially in medical domain. In contrast, most current applications of AI in medical domain tend to focus on narrowly defined tasks utilizing a single data modality and typically rely on isolated snapshots for assessments, failing to capture the continuous nature of health as an ongoing state. Therefore, in the future, development of effective multi-modal biomedical AI models will be an important research topic. My previous research has delved deeply into the fusion of multi-modal medical images, establishing a strong foundation for the development of multi-modal AI models that extend beyond images. Additionally, the increasing accessibility of biomedical data from extensive biobanks, electronic health records, medical imaging, wearable and ambient biosensors has created an ideal environment for the advancement of multi-modal AI models. Building upon these advancements, I aim to develop multi-modal AI models capable of seamlessly incorporating data across diverse modalities, including data from biosensors, genetics, imaging, clinical records, and environmental factors. The goal of these models is to facilitate broad applications such as personalized medicine and precision prediction of disease progression.

REFERENCES

[1] 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.

[2] 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.

[3] T. N. Tombaugh and N. J. McIntyre, "The mini-mental state examination: a comprehensive review," *Journal of the American Geriatrics Society*, vol. 40, no. 9, pp. 922–935, 1992.

[4] J. K. Kueper, M. Speechley, and M. Montero-Odasso, "The alzheimer's disease assessment scale–cognitive subscale (adas-cog): modifications and responsiveness in pre-dementia populations. a narrative review," *Journal of Alzheimer's Disease*, vol. 63, no. 2, pp. 423–444, 2018.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[6] A. M. Turing, *Computing machinery and intelligence.* Springer, 2009.

[7] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

[10] W. James, *The principles of psychology.* Cosimo, Inc., 2007, vol. 1.

[11] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] J. You, J. Leskovec, K. He, and S. Xie, “Graph structure of neural networks,” in *International Conference on Machine Learning.* PMLR, 2020, pp. 10 881–10 891.

[13] R. E. Passingham, K. E. Stephan, and R. Kötter, “The anatomical basis of functional localization in the cortex,” *Nature Reviews Neuroscience*, vol. 3, no. 8, pp. 606–616, 2002.

[14] D. Zhu, D. Zhang, C. Faraco, K. Li, F. Deng, H. Chen, X. Jiang, L. Guo, L. S. Miller, and T. Liu, “Discovering dense and consistent landmarks in the brain,” in *Information Processing in Medical Imaging: 22nd International Conference, IPMI 2011, Kloster Irsee, Germany, July 3-8, 2011. Proceedings 22.* Springer, 2011, pp. 97–110.

[15] D. Zhu, K. Li, C. C. Faraco, F. Deng, D. Zhang, L. Guo, L. S. Miller, and T. Liu, “Optimization of functional brain rois via maximization of consistency of structural connectivity profiles,” *NeuroImage*, vol. 59, no. 2, pp. 1382–1393, 2012.

[16] D. Zhu, K. Li, L. Guo, X. Jiang, T. Zhang, D. Zhang, H. Chen, F. Deng, C. Faraco, C. Jin, *et al.*, “Dicccol: dense individualized and common

connectivity-based cortical landmarks," *Cerebral cortex*, vol. 23, no. 4, pp. 786–800, 2013.

[17] M. A. Koch, D. G. Norris, and M. Hund-Georgiadis, "An investigation of functional and anatomical connectivity using magnetic resonance imaging," *Neuroimage*, vol. 16, no. 1, pp. 241–250, 2002.

[18] P. Skudlarski, K. Jagannathan, V. D. Calhoun, M. Hampson, B. A. Skudlarska, and G. Pearlson, "Measuring brain connectivity: diffusion tensor imaging validates resting state temporal correlations," *Neuroimage*, vol. 43, no. 3, pp. 554–561, 2008.

[19] C. J. Honey, O. Sporns, L. Cammoun, X. Gigandet, J.-P. Thiran, R. Meuli, and P. Hagmann, "Predicting human resting-state functional connectivity from structural connectivity," *Proceedings of the National Academy of Sciences*, vol. 106, no. 6, pp. 2035–2040, 2009.

[20] G. Gong, Y. He, L. Concha, C. Lebel, D. W. Gross, A. C. Evans, and C. Beaulieu, "Mapping anatomical connectivity patterns of human cerebral cortex using in vivo diffusion tensor imaging tractography," *Cerebral cortex*, vol. 19, no. 3, pp. 524–536, 2009.

[21] Z. Wang, L. M. Chen, L. Négyessy, R. M. Friedman, A. Mishra, J. C. Gore, and A. W. Roe, "The relationship of anatomical and functional connectivity to resting-state connectivity in primate somatosensory cortex," *Neuron*, vol. 78, no. 6, pp. 1116–1126, 2013.

[22] V. Pernice, B. Staude, S. Cardanobile, and S. Rotter, "How structure determines correlations in neuronal networks," *PLoS computational biology*, vol. 7, no. 5, p. e1002059, 2011.

[23] M. D. Greicius, K. Supekar, V. Menon, and R. F. Dougherty, “Resting-state functional connectivity reflects structural connectivity in the default mode network,” *Cerebral cortex*, vol. 19, no. 1, pp. 72–78, 2009.

[24] B. Mišić, R. F. Betzel, M. A. De Reus, M. P. Van Den Heuvel, M. G. Berman, A. R. McIntosh, and O. Sporns, “Network-level structure-function relationships in human neocortex,” *Cerebral Cortex*, vol. 26, no. 7, pp. 3285–3296, 2016.

[25] S. Achard, R. Salvador, B. Whitcher, J. Suckling, and E. Bullmore, “A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs,” *Journal of Neuroscience*, vol. 26, no. 1, pp. 63–72, 2006.

[26] Y. Iturria-Medina, R. C. Sotero, E. J. Canales-Rodríguez, Y. Alemán-Gómez, and L. Melie-García, “Studying the human brain anatomical network via diffusion-weighted mri and graph theory,” *Neuroimage*, vol. 40, no. 3, pp. 1064–1076, 2008.

[27] O. Sporns and J. D. Zwi, “The small world of the cerebral cortex,” *Neuroinformatics*, vol. 2, pp. 145–162, 2004.

[28] G. Zamora-López, Y. Chen, G. Deco, M. L. Kringelbach, and C. Zhou, “Functional complexity emerging from anatomical constraints in the brain: the significance of network modularity and rich-clubs,” *Scientific reports*, vol. 6, no. 1, p. 38424, 2016.

[29] I. Diez, P. Bonifazi, I. Escudero, B. Mateos, M. A. Muñoz, S. Stramaglia, and J. M. Cortes, “A novel brain partition highlights the modular skeleton shared by structure and function,” *Scientific reports*, vol. 5, no. 1, p. 10532, 2015.

[30] M. P. Van Den Heuvel and O. Sporns, “Rich-club organization of the human connectome,” *Journal of Neuroscience*, vol. 31, no. 44, pp. 15 775–15 786, 2011.

[31] M. P. Van Den Heuvel, R. S. Kahn, J. Goñi, and O. Sporns, "High-cost, high-capacity backbone for global brain communication," *Proceedings of the National Academy of Sciences*, vol. 109, no. 28, pp. 11 372–11 377, 2012.

[32] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[34] L. Sun, Z. Fan, X. Ding, Y. Huang, and J. Paisley, "Region-of-interest undersampled mri reconstruction: a deep convolutional neural network approach," *Magnetic resonance imaging*, vol. 63, pp. 185–192, 2019.

[35] L. Wang, Y. Gao, F. Shi, G. Li, J. H. Gilmore, W. Lin, and D. Shen, "Links: Learning-based multi-source integration framework for segmentation of infant brain images," *NeuroImage*, vol. 108, pp. 160–172, 2015.

[36] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. Snead, I. A. Cree, and N. M. Rajpoot, "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1196–1206, 2016.

[37] H. R. Roth, L. Lu, J. Liu, J. Yao, A. Seff, K. Cherry, L. Kim, and R. M. Summers, "Improving computer-aided detection using convolutional neural networks and random view aggregation," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1170–1181, 2015.

[38] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[39] S. I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, and D. Rueckert, "Metric learning with spectral graph convolutions on brain connectivity networks," *NeuroImage*, vol. 169, pp. 431–442, 2018.

[40] A. Kazi, S. Shekarforoush, S. Arvind Krishna, H. Burwinkel, G. Vivar, K. Kortüm, S.-A. Ahmadi, S. Albarqouni, and N. Navab, "Inceptiongcn: receptive field aware graph convolutional network for disease prediction," in *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26.* Springer, 2019, pp. 73–85.

[41] L. Zhang, L. Wang, J. Gao, S. L. Risacher, J. Yan, G. Li, T. Liu, D. Zhu, A. D. N. Initiative, *et al.*, "Deep fusion of brain structure-function in mild cognitive impairment," *Medical image analysis*, vol. 72, p. 102082, 2021.

[42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[43] L. Zhao and L. Akoglu, "Pairnorm: Tackling oversmoothing in gnns," *arXiv preprint arXiv:1909.12223*, 2019.

[44] R. C. Petersen, P. S. Aisen, L. A. Beckett, M. C. Donohue, A. C. Gamst, D. J. Harvey, C. Jack Jr, W. J. Jagust, L. M. Shaw, A. W. Toga, *et al.*, "Alzheimer's disease neuroimaging initiative (adni) clinical characterization," *Neurology*, vol. 74, no. 3, pp. 201–209, 2010.

[45] K. Batista-García-Ramó and C. I. Fernández-Verdecia, "What we know about the brain structure–function relationship," *Behavioral Sciences*, vol. 8, no. 4, p. 39, 2018.

[46] D. Zhu, K. Li, D. P. Terry, A. N. Puente, L. Wang, D. Shen, L. S. Miller, and T. Liu, "Connectome-scale assessments of structural and functional connectivity in mci," *Human brain mapping*, vol. 35, no. 7, pp. 2911–2923, 2014.

[47] 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.

[48] N. Toussaint, J.-C. Souplet, and P. Fillard, "Medinria: medical image navigation and research tool by inria," in *Proc. of MICCAI'07 Workshop on Interaction in medical image analysis and visualization*, 2007.

[49] B. Fischl, "Freesurfer," *Neuroimage*, vol. 62, no. 2, pp. 774–781, 2012.

[50] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

[51] G. Marrelec, A. Krainik, H. Duffau, M. Pélégrini-Issac, S. Lehéricy, J. Doyon, and H. Benali, "Partial correlation for functional brain interactivity investigation in functional mri," *Neuroimage*, vol. 32, no. 1, pp. 228–237, 2006.

[52] E. Challis, P. Hurley, L. Serra, M. Bozzali, S. Oliver, and M. Cercignani, "Gaussian process classification of alzheimer's disease and mild cognitive impairment from resting-state fmri," *NeuroImage*, vol. 112, pp. 232–243, 2015.

[53] D. Curran-Everett, "Explorations in statistics: the log transformation," *Advances in physiology education*, vol. 42, no. 2, pp. 343–347, 2018.

[54] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[55] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[56] L. Zhang, A. Zaman, L. Wang, J. Yan, and D. Zhu, "A cascaded multi-modality analysis in mild cognitive impairment," in *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 10.* Springer, 2019, pp. 557–565.

[57] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: uses and interpretations," *Neuroimage*, vol. 52, no. 3, pp. 1059–1069, 2010.

[58] D. Gusfield, "Computing the strength of a graph," *SIAM Journal on Computing*, vol. 20, no. 4, pp. 639–654, 1991.

[59] F. Monti, K. Otness, and M. M. Bronstein, "Motifnet: a motif-based graph convolutional network for directed graphs," in *2018 IEEE data science workshop (DSW)*. IEEE, 2018, pp. 225–228.

[60] Y. Ma, J. Hao, Y. Yang, H. Li, J. Jin, and G. Chen, "Spectral-based graph convolutional network for directed graphs," *arXiv preprint arXiv:1907.08990*, 2019.

[61] Z. Tong, Y. Liang, C. Sun, D. S. Rosenblum, and A. Lim, "Directed graph convolutional network," *arXiv preprint arXiv:2004.13970*, 2020.

[62] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3681–3688.

[63] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial intelligence*, vol. 267, pp. 1–38, 2019.

[64] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[65] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International conference on machine learning*. PMLR, 2017, pp. 1885–1894.

[66] J. S. Damoiseaux and M. D. Greicius, "Greater than the sum of its parts: a review of studies combining structural connectivity and resting-state functional connectivity," *Brain structure and function*, vol. 213, pp. 525–533, 2009.

[67] C.-Y. Wee, P.-T. Yap, D. Zhang, K. Denny, J. N. Browndyke, G. G. Potter, K. A. Welsh-Bohmer, L. Wang, and D. Shen, “Identification of mci individuals using structural and functional connectivity networks,” *Neuroimage*, vol. 59, no. 3, pp. 2045–2056, 2012.

[68] H.-J. Park and K. Friston, “Structural and functional brain networks: from connections to cognition,” *Science*, vol. 342, no. 6158, p. 1238411, 2013.

[69] J. Sui, T. Adali, Q. Yu, J. Chen, and V. D. Calhoun, “A review of multivariate methods for multimodal fusion of brain imaging data,” *Journal of neuroscience methods*, vol. 204, no. 1, pp. 68–81, 2012.

[70] J. Sui, R. Huster, Q. Yu, J. M. Segall, and V. D. Calhoun, “Function–structure associations of the brain: evidence from multimodal connectivity and covariance studies,” *Neuroimage*, vol. 102, pp. 11–23, 2014.

[71] K. Uludağ and A. Roebroeck, “General overview on the merits of multimodal neuroimaging data fusion,” *Neuroimage*, vol. 102, pp. 3–10, 2014.

[72] D. Zhu, T. Zhang, X. Jiang, X. Hu, H. Chen, N. Yang, J. Lv, J. Han, L. Guo, and T. Liu, “Fusing dti and fmri data: a survey of methods and applications,” *NeuroImage*, vol. 102, pp. 184–191, 2014.

[73] S. Mori, B. J. Crain, V. P. Chacko, and P. C. Van Zijl, “Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging,” *Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society*, vol. 45, no. 2, pp. 265–269, 1999.

[74] K. Supekar, V. Menon, D. Rubin, M. Musen, and M. D. Greicius, “Network analysis of intrinsic functional brain connectivity in alzheimer’s disease,” *PLoS computational biology*, vol. 4, no. 6, p. e1000100, 2008.

[75] T. Xie and Y. He, “Mapping the alzheimer’s brain with connectomics,” *Frontiers in psychiatry*, vol. 2, p. 18024, 2012.

[76] G. Chen, H.-Y. Zhang, C. Xie, G. Chen, Z.-J. Zhang, G.-J. Teng, and S.-J. Li, "Modular reorganization of brain resting state networks and its independent validation in alzheimer's disease patients," *Frontiers in human neuroscience*, vol. 7, p. 456, 2013.

[77] M. Daianu, N. Jahanshad, T. M. Nir, A. W. Toga, C. R. Jack Jr, M. W. Weiner, and P. M. Thompson, for the Alzheimer's Disease Neuroimaging Initiative, "Breakdown of brain connectivity between normal aging and alzheimer's disease: a structural k-core network analysis," *Brain connectivity*, vol. 3, no. 4, pp. 407–422, 2013.

[78] N. Franzmeier and M. Dyrba, "Functional brain network architecture may route progression of alzheimer's disease pathology," *Brain*, vol. 140, no. 12, pp. 3077–3080, 2017.

[79] Z. Dai, Q. Lin, T. Li, X. Wang, H. Yuan, X. Yu, Y. He, and H. Wang, "Disrupted structural and functional brain networks in alzheimer's disease," *Neurobiology of aging*, vol. 75, pp. 71–82, 2019.

[80] T. Schonberg, P. Pianka, T. Hendler, O. Pasternak, and Y. Assaf, "Characterization of displaced white matter by brain tumors using combined dti and fmri," *Neuroimage*, vol. 30, no. 4, pp. 1100–1111, 2006.

[81] P. Hagmann, O. Sporns, N. Madan, L. Cammoun, R. Pienaar, V. J. Wedeen, R. Meuli, J.-P. Thiran, and P. Grant, "White matter maturation reshapes structural connectivity in the late developing human brain," *Proceedings of the National Academy of Sciences*, vol. 107, no. 44, pp. 19 067–19 072, 2010.

[82] B. W. Fling, Y. Kwak, S. J. Peltier, and R. D. Seidler, "Differential relationships between transcallosal structural and functional connectivity in young and older adults," *Neurobiology of aging*, vol. 33, no. 10, pp. 2521–2526, 2012.

[83] B. Jie, D. Zhang, W. Gao, Q. Wang, C.-Y. Wee, and D. Shen, "Integration of network topological and connectivity properties for neuroimaging classification," *IEEE transactions on biomedical engineering*, vol. 61, no. 2, pp. 576–589, 2013.

[84] H.-I. Suk, S.-W. Lee, D. Shen, A. D. N. Initiative, *et al.*, "Hierarchical feature representation and multimodal fusion with deep learning for ad/mci diagnosis," *NeuroImage*, vol. 101, pp. 569–582, 2014.

[85] X. Meng, R. Jiang, D. Lin, J. Bustillo, T. Jones, J. Chen, Q. Yu, Y. Du, Y. Zhang, T. Jiang, *et al.*, "Predicting individualized clinical measures by a generalized prediction framework and multimodal fusion of mri data," *Neuroimage*, vol. 145, pp. 218–229, 2017.

[86] M. Liu, D. Cheng, K. Wang, Y. Wang, and A. D. N. Initiative, "Multi-modality cascaded convolutional neural networks for alzheimer's disease diagnosis," *Neuroinformatics*, vol. 16, pp. 295–308, 2018.

[87] S. M. Plis, M. F. Amin, A. Chekroud, D. Hjelm, E. Damaraju, H. J. Lee, J. R. Bustillo, K. Cho, G. D. Pearlson, and V. D. Calhoun, "Reading the (functional) writing on the (structural) wall: Multimodal fusion of brain structure and function via a deep neural network based translation approach reveals novel impairments in schizophrenia," *NeuroImage*, vol. 181, pp. 734–747, 2018.

[88] K. Li, L. Guo, D. Zhu, X. Hu, J. Han, and T. Liu, "Individual functional roi optimization via maximization of group-wise consistency of structural and functional profiles," *Neuroinformatics*, vol. 10, pp. 225–242, 2012.

[89] S. M. Daselaar, V. Iyengar, S. W. Davis, K. Eklund, S. M. Hayes, and R. E. Cabeza, "Less wiring, more firing: low-performing older adults compensate for impaired white matter with greater neural activity," *Cerebral cortex*, vol. 25, no. 4, pp. 983–990, 2015.

[90] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[91] F. Monti, M. Bronstein, and X. Bresson, “Geometric matrix completion with recurrent multi-graph neural networks,” *Advances in neural information processing systems*, vol. 30, 2017.

[92] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[93] G. Tononi, O. Sporns, and G. M. Edelman, “A measure for brain complexity: relating functional segregation and integration in the nervous system.” *Proceedings of the National Academy of Sciences*, vol. 91, no. 11, pp. 5033–5037, 1994.

[94] E. Bullmore and O. Sporns, “Complex brain networks: graph theoretical analysis of structural and functional systems,” *Nature reviews neuroscience*, vol. 10, no. 3, pp. 186–198, 2009.

[95] A. Fornito, A. Zalesky, and M. Breakspear, “The connectomics of brain disorders,” *Nature Reviews Neuroscience*, vol. 16, no. 3, pp. 159–172, 2015.

[96] P. Fries, “A mechanism for cognitive dynamics: neuronal communication through neuronal coherence,” *Trends in cognitive sciences*, vol. 9, no. 10, pp. 474–480, 2005.

[97] S. E. Petersen and O. Sporns, “Brain networks and cognitive architectures,” *Neuron*, vol. 88, no. 1, pp. 207–219, 2015.

[98] C. W. Lynn and D. S. Bassett, “The physics of brain network structure, function and control,” *Nature Reviews Physics*, vol. 1, no. 5, pp. 318–332, 2019.

[99] S. Vieira, W. H. Pinaya, and A. Mechelli, “Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications,” *Neuroscience & Biobehavioral Reviews*, vol. 74, pp. 58–75, 2017.

[100] J. L. Guo and V. M. Lee, “Cell-to-cell transmission of pathogenic proteins in neurodegenerative diseases,” *Nature medicine*, vol. 20, no. 2, pp. 130–138, 2014.

[101] J. S. Phillips, F. Da Re, D. J. Irwin, C. T. McMillan, S. N. Vaishnavi, S. X. Xie, E. B. Lee, P. A. Cook, J. C. Gee, L. M. Shaw, *et al.*, “Longitudinal progression of grey matter atrophy in non-amnestic alzheimer’s disease,” *Brain*, vol. 142, no. 6, pp. 1701–1722, 2019.

[102] J. Hlinka, M. Paluš, M. Vejmelka, D. Mantini, and M. Corbetta, “Functional connectivity in resting-state fmri: is linear correlation sufficient?” *Neuroimage*, vol. 54, no. 3, pp. 2218–2225, 2011.

[103] S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich, “Network modelling methods for fmri,” *Neuroimage*, vol. 54, no. 2, pp. 875–891, 2011.

[104] E. Xing, M. Jordan, S. J. Russell, and A. Ng, “Distance metric learning with application to clustering with side-information,” *Advances in neural information processing systems*, vol. 15, 2002.

[105] L. Zhang, L. Wang, and D. Zhu, “Jointly analyzing alzheimer’s disease related structure-function using deep cross-model attention network,” in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 563–567.

[106] 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.

[107] B. Cheng, M. Liu, D. Zhang, B. C. Munsell, and D. Shen, “Domain transfer learning for mci conversion prediction,” *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 7, pp. 1805–1817, 2015.

[108] M. Dyrba, M. Grothe, T. Kirste, and S. J. Teipel, “Multimodal analysis of functional and structural disconnection in a lzheimer’s disease using multiple kernel svm,” *Human brain mapping*, vol. 36, no. 6, pp. 2118–2131, 2015.

[109] M. Liu, D. Zhang, E. Adeli, and D. Shen, “Inherent structure-based multiview learning with multitemplate feature representation for alzheimer’s disease diagnosis,” *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1473–1482, 2015.

[110] C. Möller, Y. A. Pijnenburg, W. M. van der Flier, A. Versteeg, B. Tijms, J. C. de Munck, A. Hafkemeijer, S. A. Rombouts, J. van der Grond, J. van Swieten, *et al.*, “Alzheimer disease and behavioral variant frontotemporal dementia: automatic classification based on cortical atrophy for single-subject diagnosis,” *Radiology*, vol. 279, no. 3, pp. 838–848, 2016.

[111] W. Shao, Y. Peng, C. Zu, M. Wang, D. Zhang, A. D. N. Initiative, *et al.*, “Hypergraph based multi-task feature selection for multimodal classification of alzheimer’s disease,” *Computerized Medical Imaging and Graphics*, vol. 80, p. 101663, 2020.

[112] K. Aderghal, A. Khvostikov, A. Krylov, J. Benois-Pineau, K. Afdel, and G. Catheline, “Classification of alzheimer disease on imaging modalities with deep cnns using cross-modal transfer learning,” in *2018 IEEE 31st international symposium on computer-based medical systems (CBMS)*. IEEE, 2018, pp. 345–350.

[113] Y. Huang, J. Xu, Y. Zhou, T. Tong, X. Zhuang, and A. D. N. I. (ADNI), "Diagnosis of alzheimer's disease via multi-modality 3d convolutional neural network," *Frontiers in neuroscience*, vol. 13, p. 509, 2019.

[114] S.-Y. Tsai, "Reproducibility of structural brain connectivity and network metrics using probabilistic diffusion tractography," *Scientific reports*, vol. 8, no. 1, p. 11562, 2018.

[115] G. W. Van Hoesen, J. Parvizi, and C.-C. Chu, "Orbitofrontal cortex pathology in alzheimer's disease," *Cerebral Cortex*, vol. 10, no. 3, pp. 243–251, 2000.

[116] Y. Liu, C. Yu, X. Zhang, J. Liu, Y. Duan, A. F. Alexander-Bloch, B. Liu, T. Jiang, and E. Bullmore, "Impaired long distance functional connectivity and weighted network architecture in alzheimer's disease," *Cerebral Cortex*, vol. 24, no. 6, pp. 1422–1435, 2014.

[117] W. Henneman, J. Sluimer, J. Barnes, W. Van Der Flier, I. Sluimer, N. Fox, P. Scheltens, H. Vrenken, and F. Barkhof, "Hippocampal atrophy rates in alzheimer disease: added value over whole brain volume measures," *Neurology*, vol. 72, no. 11, pp. 999–1007, 2009.

[118] G. Karas, P. Scheltens, S. A. Rombouts, P. J. Visser, R. A. van Schijndel, N. C. Fox, and F. Barkhof, "Global and local gray matter loss in mild cognitive impairment and alzheimer's disease," *Neuroimage*, vol. 23, no. 2, pp. 708–716, 2004.

[119] S. Li, F. Pu, F. Shi, S. Xie, Y. Wang, and T. Jiang, "Regional white matter decreases in alzheimer's disease using optimized voxel-based morphometry," *Acta radiologica*, vol. 49, no. 1, pp. 84–90, 2008.

[120] M. D. Greicius, G. Srivastava, A. L. Reiss, and V. Menon, "Default-mode network activity distinguishes alzheimer's disease from healthy aging: evidence

from functional mri," *Proceedings of the National Academy of Sciences*, vol. 101, no. 13, pp. 4637–4642, 2004.

[121] J. Ashburner and K. J. Friston, "Voxel-based morphometry—the methods," *Neuroimage*, vol. 11, no. 6, pp. 805–821, 2000.

[122] S. M. Smith, M. Jenkinson, H. Johansen-Berg, D. Rueckert, T. E. Nichols, C. E. Mackay, K. E. Watkins, O. Ciccarelli, M. Z. Cader, P. M. Matthews, *et al.*, "Tract-based spatial statistics: voxelwise analysis of multi-subject diffusion data," *Neuroimage*, vol. 31, no. 4, pp. 1487–1505, 2006.

[123] L. Wang, L. Zhang, and D. Zhu, "Learning latent structure over deep fusion model of mild cognitive impairment," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 1039–1043.

[124] P. S. Aisen, J. Cummings, C. R. Jack, J. C. Morris, R. Sperling, L. Frölich, R. W. Jones, S. A. Dowsett, B. R. Matthews, J. Raskin, *et al.*, "On the path to 2025: understanding the alzheimer's disease continuum," *Alzheimer's research & therapy*, vol. 9, pp. 1–10, 2017.

[125] M. Ten Kate, E. Dicks, P. J. Visser, W. M. van der Flier, C. E. Teunissen, F. Barkhof, P. Scheltens, B. M. Tijms, and A. D. N. Initiative, "Atrophy subtypes in prodromal alzheimer's disease are associated with cognitive decline," *Brain*, vol. 141, no. 12, pp. 3443–3456, 2018.

[126] S. J. Vos, F. Verhey, L. Frölich, J. Kornhuber, J. Wiltfang, W. Maier, O. Peters, E. Rüther, F. Nobili, S. Morbelli, *et al.*, "Prevalence and prognosis of alzheimer's disease at the mild cognitive impairment stage," *Brain*, vol. 138, no. 5, pp. 1327–1338, 2015.

[127] B. Lei, Y. Zhao, Z. Huang, X. Hao, F. Zhou, A. Elazab, J. Qin, and H. Lei, "Adaptive sparse learning using multi-template for neurodegenerative disease diagnosis," *Medical Image Analysis*, vol. 61, p. 101632, 2020.

[128] G. B. Frisoni, N. C. Fox, C. R. Jack Jr, P. Scheltens, and P. M. Thompson, “The clinical use of structural mri in alzheimer disease,” *Nature Reviews Neurology*, vol. 6, no. 2, pp. 67–77, 2010.

[129] A. Mouiha, S. Duchesne, A. D. N. Initiative, *et al.*, “Toward a dynamic biomarker model in alzheimer’s disease,” *Journal of Alzheimer’s Disease*, vol. 30, no. 1, pp. 91–100, 2012.

[130] A. L. Young, N. P. Oxtoby, P. Daga, D. M. Cash, N. C. Fox, S. Ourselin, J. M. Schott, and D. C. Alexander, “A data-driven model of biomarker changes in sporadic alzheimer’s disease,” *Brain*, vol. 137, no. 9, pp. 2564–2577, 2014.

[131] D. Li, S. Iddi, W. K. Thompson, M. C. Donohue, and A. D. N. Initiative, “Bayesian latent time joint mixed effect models for multicohort longitudinal data,” *Statistical methods in medical research*, vol. 28, no. 3, pp. 835–845, 2019.

[132] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, “A survey of text representation and embedding techniques in nlp,” *IEEE Access*, 2023.

[133] L. Zhang, L. Zhao, D. Liu, Z. Wu, X. Wang, T. Liu, and D. Zhu, “Cortex2vector: anatomical embedding of cortical folding patterns,” *Cerebral Cortex*, vol. 33, no. 10, pp. 5851–5862, 2023.

[134] N. Amoroso, D. Diacono, A. Fanizzi, M. La Rocca, A. Monaco, A. Lombardi, C. Guaragnella, R. Bellotti, S. Tangaro, A. D. N. Initiative, *et al.*, “Deep learning reveals alzheimer’s disease onset in mci subjects: results from an international challenge,” *Journal of neuroscience methods*, vol. 302, pp. 3–9, 2018.

[135] T. Zhou, K.-H. Thung, X. Zhu, and D. Shen, “Effective feature learning and fusion of multimodality data using stage-wise deep neural network for dementia diagnosis,” *Human brain mapping*, vol. 40, no. 3, pp. 1001–1016, 2019.

[136] L. Brand, K. Nichols, H. Wang, L. Shen, and H. Huang, "Joint multi-modal longitudinal regression and classification for alzheimer's disease prediction," *IEEE transactions on medical imaging*, vol. 39, no. 6, pp. 1845–1855, 2019.

[137] A. Puente-Castro, E. Fernandez-Blanco, A. Pazos, and C. R. Munteanu, "Automatic assessment of alzheimer's disease diagnosis based on deep learning techniques," *Computers in biology and medicine*, vol. 120, p. 103764, 2020.

[138] J. Liu, M. Li, Y. Luo, S. Yang, W. Li, and Y. Bi, "Alzheimer's disease detection using depthwise separable convolutional neural networks," *Computer Methods and Programs in Biomedicine*, vol. 203, p. 106032, 2021.

[139] Z. Xu, H. Deng, J. Liu, and Y. Yang, "Diagnosis of alzheimer's disease based on the modified tresnet," *Electronics*, vol. 10, no. 16, p. 1908, 2021.

[140] W. Lin, Q. Gao, M. Du, W. Chen, and T. Tong, "Multiclass diagnosis of stages of alzheimer's disease using linear discriminant analysis scoring for multimodal data," *Computers in Biology and Medicine*, vol. 134, p. 104478, 2021.

[141] A. W. Mulyadi, W. Jung, K. Oh, J. S. Yoon, K. H. Lee, and H.-I. Suk, "Estimating explainable alzheimer's disease likelihood map via clinically-guided prototype learning," *NeuroImage*, vol. 273, p. 120073, 2023.

[142] M. Sudharsan and G. Thailambal, "Alzheimer's disease prediction using machine learning techniques and principal component analysis (pca)," *Materials Today: Proceedings*, vol. 81, pp. 182–190, 2023.

[143] L. Zhang, L. Wang, and D. Zhu, "Recovering brain structural connectivity from functional connectivity via multi-gcn based generative adversarial network," 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. 53–61.

[144] L. Zhang, L. Wang, D. Zhu, A. D. N. Initiative, *et al.*, “Predicting brain structural network using functional connectivity,” *Medical image analysis*, vol. 79, p. 102463, 2022.

[145] X. Zhang, L. Guo, X. Li, T. Zhang, D. Zhu, K. Li, H. Chen, J. Lv, C. Jin, Q. Zhao, *et al.*, “Characterization of task-free and task-performance brain states via functional connectome patterns,” *Medical image analysis*, vol. 17, no. 8, pp. 1106–1122, 2013.

[146] X. He, D. Cai, and P. Niyogi, “Laplacian score for feature selection,” *Advances in neural information processing systems*, vol. 18, 2005.

[147] W. Chen, T.-Y. Liu, Y. Lan, Z.-M. Ma, and H. Li, “Ranking measures and loss functions in learning to rank,” *Advances in Neural Information Processing Systems*, vol. 22, 2009.

[148] D. Cossock and T. Zhang, “Statistical analysis of bayes optimal subset ranking,” *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 5140–5154, 2008.

[149] P. Li, Q. Wu, and C. Burges, “Mcrank: Learning to rank using multiple classification and gradient boosting,” *Advances in neural information processing systems*, vol. 20, 2007.

[150] R. Herbrich, T. Graepel, and K. Obermayer, “Large margin rank boundaries for ordinal regression,” 2000.

[151] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, “An efficient boosting algorithm for combining preferences,” *Journal of machine learning research*, vol. 4, no. Nov, pp. 933–969, 2003.

[152] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, “Learning to rank: from pairwise approach to listwise approach,” in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 129–136.

[153] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, "Listwise approach to learning to rank: theory and algorithm," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1192–1199.

[154] L. Wang, L. Zhang, and D. Zhu, "Accessing latent connectome of mild cognitive impairment via discriminant structure learning," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 164–168.

[155] F. Palesi, G. Castellazzi, L. Casiraghi, E. Sinforiani, P. Vitali, C. A. Gandini Wheeler-Kingshott, and E. D'Angelo, "Exploring patterns of alteration in alzheimer's disease brain networks: a combined structural and functional connectomics analysis," *Frontiers in neuroscience*, vol. 10, p. 380, 2016.

[156] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[157] L. Zhao, L. Zhang, Z. Wu, Y. Chen, H. Dai, X. Yu, Z. Liu, T. Zhang, X. Hu, X. Jiang, *et al.*, "When brain-inspired ai meets agi," *Meta-Radiology*, p. 100005, 2023.

[158] X. Li, L. Zhang, Z. Wu, Z. Liu, L. Zhao, Y. Yuan, J. Liu, G. Li, D. Zhu, P. Yan, *et al.*, "Artificial general intelligence for medical imaging," *arXiv preprint arXiv:2306.05480*, 2023.

[159] Z. Xiao, Y. Chen, L. Zhang, J. Yao, Z. Wu, X. Yu, Y. Pan, L. Zhao, C. Ma, X. Liu, *et al.*, "Instruction-vit: Multi-modal prompts for instruction learning in vit," *arXiv preprint arXiv:2305.00201*, 2023.

[160] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from

natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[161] K. Zilles, E. Armstrong, A. Schleicher, and H.-J. Kretschmann, "The human pattern of gyrification in the cerebral cortex," *Anatomy and embryology*, vol. 179, pp. 173–179, 1988.

[162] G. Roth and U. Dicke, "Evolution of the brain and intelligence," *Trends in cognitive sciences*, vol. 9, no. 5, pp. 250–257, 2005.

[163] C. C. Hilgetag and H. Barbas, "Role of mechanical factors in the morphology of the primate cerebral cortex," *PLoS computational biology*, vol. 2, no. 3, p. e22, 2006.

[164] B. Fischl, N. Rajendran, E. Busa, J. Augustinack, O. Hinds, B. T. Yeo, H. Mohlberg, K. Amunts, and K. Zilles, "Cortical folding patterns and predicting cytoarchitecture," *Cerebral cortex*, vol. 18, no. 8, pp. 1973–1980, 2008.

[165] C. J. Honey, J.-P. Thivierge, and O. Sporns, "Can structure predict function in the human brain?" *Neuroimage*, vol. 52, no. 3, pp. 766–776, 2010.

[166] P. M. Thompson, K. M. Hayashi, E. R. Sowell, N. Gogtay, J. N. Giedd, J. L. Rapoport, G. I. De Zubicaray, A. L. Janke, S. E. Rose, J. Semple, *et al.*, "Mapping cortical change in alzheimer's disease, brain development, and schizophrenia," *Neuroimage*, vol. 23, pp. S2–S18, 2004.

[167] P. C. Sallet, H. Elkis, T. M. Alves, J. R. Oliveira, E. Sassi, C. C. de Castro, G. F. Busatto, and W. F. Gattaz, "Reduced cortical folding in schizophrenia: an mri morphometric study," *American Journal of Psychiatry*, vol. 160, no. 9, pp. 1606–1613, 2003.

[168] A. Y. Hardan, R. J. Jou, M. S. Keshavan, R. Varma, and N. J. Minshew, "Increased frontal cortical folding in autism: a preliminary mri study," *Psychiatry Research: Neuroimaging*, vol. 131, no. 3, pp. 263–268, 2004.

[169] C. W. Nordahl, D. Dierker, I. Mostafavi, C. M. Schumann, S. M. Rivera, D. G. Amaral, and D. C. Van Essen, "Cortical folding abnormalities in autism revealed by surface-based morphometry," *Journal of Neuroscience*, vol. 27, no. 43, pp. 11 725–11 735, 2007.

[170] K. Li, L. Guo, G. Li, J. Nie, C. Faraco, G. Cui, Q. Zhao, L. S. Miller, and T. Liu, "Gyral folding pattern analysis via surface profiling," *NeuroImage*, vol. 52, no. 4, pp. 1202–1214, 2010.

[171] H. Chen, Y. Li, F. Ge, G. Li, D. Shen, and T. Liu, "Gyral net: A new representation of cortical folding organization," *Medical image analysis*, vol. 42, pp. 14–25, 2017.

[172] X. Li, H. Chen, T. Zhang, X. Yu, X. Jiang, K. Li, L. Li, M. J. Razavi, X. Wang, X. Hu, *et al.*, "Commonly preserved and species-specific gyral folding patterns across primate brains," *Brain structure and function*, vol. 222, pp. 2127–2141, 2017.

[173] F. Ge, X. Li, M. J. Razavi, H. Chen, T. Zhang, S. Zhang, L. Guo, X. Hu, X. Wang, and T. Liu, "Denser growing fiber connections induce 3-hinge gyral folding," *Cerebral Cortex*, vol. 28, no. 3, pp. 1064–1075, 2018.

[174] T. Zhang, X. Li, X. Jiang, F. Ge, S. Zhang, L. Zhao, H. Liu, Y. Huang, X. Wang, J. Yang, *et al.*, "Cortical 3-hinges could serve as hubs in cortico-cortical connective network," *Brain imaging and behavior*, vol. 14, pp. 2512–2529, 2020.

[175] T. Zhang, Y. Huang, L. Zhao, Z. He, X. Jiang, L. Guo, X. Hu, and T. Liu, "Identifying cross-individual correspondences of 3-hinge gyri," *Medical image analysis*, vol. 63, p. 101700, 2020.

[176] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[177] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[178] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek, "Detecting formal thought disorder by deep contextualized word representations," *Psychiatry Research*, vol. 304, p. 114135, 2021.

[179] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[180] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[181] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[182] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "graph2vec: Learning distributed representations of graphs," *arXiv preprint arXiv:1707.05005*, 2017.

[183] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

[184] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[185] H. Chen, B. Perozzi, Y. Hu, and S. Skiena, “Harp: Hierarchical representation learning for networks,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[186] B. Fischl, M. I. Sereno, and A. M. Dale, “Cortical surface-based analysis: Ii: inflation, flattening, and a surface-based coordinate system,” *Neuroimage*, vol. 9, no. 2, pp. 195–207, 1999.

[187] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, “Improving language understanding by generative pre-training,” 2018.

[188] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[189] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[190] D. C. v. Essen, “A tension-based theory of morphogenesis and compact wiring in the central nervous system,” *Nature*, vol. 385, no. 6614, pp. 313–318, 1997.

[191] G. Xu, A. K. Knutsen, K. Dikranian, C. D. Kroenke, P. V. Bayly, and L. A. Taber, “Axons pull on the brain, but tension does not drive cortical folding,” 2010.

[192] J. Nie, L. Guo, K. Li, Y. Wang, G. Chen, L. Li, H. Chen, F. Deng, X. Jiang, T. Zhang, *et al.*, “Axonal fiber terminations concentrate on gyri,” *Cerebral cortex*, vol. 22, no. 12, pp. 2831–2839, 2012.

[193] C. Reveley, A. K. Seth, C. Pierpaoli, A. C. Silva, D. Yu, R. C. Saunders, D. A. Leopold, and F. Q. Ye, “Superficial white matter fiber systems impede detection

of long-range cortical connections in diffusion mr tractography," *Proceedings of the National Academy of Sciences*, vol. 112, no. 21, pp. E2820–E2828, 2015.

[194] H. Zhang, J. Chen, F. Jiang, F. Yu, Z. Chen, J. Li, G. Chen, X. Wu, Z. Zhang, Q. Xiao, X. Wan, B. Wang, and H. Li, "Huatuogpt, towards taming language model to be a doctor," 2023.

[195] Q. C. Ziang Leng and C. Li, "Luotuo: An instruction-following chinese language model, lora tuning on llama," https://github.com/LC1332/Luotuo-Chinese-LLM, 2023.

[196] J. Wang, Y. Zhang, L. Zhang, P. Yang, X. Gao, Z. Wu, X. Dong, J. He, J. Zhuo, Q. Yang, *et al.*, "Fengshenbang 1.0: Being the foundation of chinese cognitive intelligence," *arXiv preprint arXiv:2209.02970*, 2022.

[197] G. Li, Y. Xi, J. Ding, D. Wang, Z. Luo, R. Zhang, B. Liu, C. Fan, X. Mao, and Z. Zhao, "Easy and efficient transformer: Scalable inference solution for large NLP model," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*. Hybrid: Seattle, Washington + Online: Association for Computational Linguistics, July 2022, pp. 62–68. [Online]. Available: https://aclanthology.org/2022.naacl-industry.8

[198] H. Wang, C. Liu, N. Xi, Z. Qiang, S. Zhao, B. Qin, and T. Liu, "Huatuo: Tuning llama model with chinese medical knowledge," 2023.

[199] R. Wang, Y. Duan, J. Li, P. Pang, and T. Tan, "Xrayglm: The first chinese medical multimodal model that chest radiographs summarization," https://github.com/WangRongsheng/XrayGLM, 2023.

[200] H. Wang, C. Liu, S. Zhao, B. Qin, and T. Liu, "Chatglm-med," https://github.com/SCIR-HI/Med-ChatGLM, 2023.

[201] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155*, 2022.

[202] W. B. Knox and P. Stone, "Augmenting reinforcement learning with human feedback," in *ICML 2011 Workshop on New Developments in Imitation Learning (July 2011)*, vol. 855, 2011, p. 3.

[203] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang, "Glm: General language model pretraining with autoregressive blank infilling," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 320–335.

[204] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, *et al.*, "Glm-130b: An open bilingual pre-trained model," *arXiv preprint arXiv:2210.02414*, 2022.

[205] Z. University, "Qizhengpt," https://github.com/CMKRG/QiZhenGPT, 2023.

[206] Y. Cui, Z. Yang, and X. Yao, "Efficient and effective text encoding for chinese llama and alpaca," *arXiv preprint arXiv:2304.08177*, 2023. [Online]. Available: https://arxiv.org/abs/2304.08177

[207] T. Sun, X. Zhang, Z. He, P. Li, Q. Cheng, H. Yan, X. Liu, Y. Shao, Q. Tang, X. Zhao, K. Chen, Y. Zheng, Z. Zhou, R. Li, J. Zhan, Y. Zhou, L. Li, X. Yang, L. Wu, Z. Yin, X. Huang, and X. Qiu, "Moss: Training conversational language models from synthetic data," 2023.

[208] "Chinese-llama 1 & 2 and chinese-falcon 1 & 2," https://github.com/CVI-SZU/Linly, [Accessed 26-07-2023].

[209] Z. Zhao, Y. Li, C. Hou, J. Zhao, R. Tian, W. Liu, Y. Chen, N. Sun, H. Liu, W. Mao, *et al.*, "Tencentpretrain: A scalable and flexible toolkit for pre-training models of different modalities," *arXiv preprint arXiv:2212.06385*, 2022.

[210] Y. Li, Y. Zhang, Z. Zhao, L. Shen, W. Liu, W. Mao, and H. Zhang, "Csl: A large-scale chinese scientific literature dataset," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 3917–3923.

[211] "GitHub - OpenBMB/CPM-Bee," https://github.com/OpenBMB/CPM-Bee, [Accessed 26-07-2023].

[212] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, and D. Sontag, "Large language models are zero-shot clinical information extractors," *arXiv preprint arXiv:2205.12689*, 2022.

[213] B. I. Technology, "A large-scale 7b pretraining language model developed by baichuan-inc," https://github.com/baichuan-inc/Baichuan-7B, 2023.

[214] Atomecho, "Atomgpt," https://github.com/AtomEcho/AtomGPT, 2023.

[215] A. de Wynter, X. Wang, A. Sokolov, Q. Gu, and S.-Q. Chen, "An evaluation on large language model outputs: Discourse and memorization," *arXiv preprint arXiv:2304.08637*, 2023.

[216] L. X. Xuanwei Zhang and K. Zhao, "Chatyuan: A large language model for dialogue in chinese and english," Dec. 2022. [Online]. Available: https://github.com/clue-ai/ChatYuan

[217] Y. Chen, Z. Wang, X. Xing, Z. Xu, K. Fang, S. Li, J. Wang, and X. Xu, "Bianque-1.0: Improving the "question" ability of medical chat model through finetuning with hybrid instructions and multi-turn doctor qa datasets," 2023. [Online]. Available: https://github.com/scutcyr/BianQue

[218] B. A. of Artificial Intelligence, "Aquilachat," https://model.baai.ac.cn/model-detail/100101, 2023.

[219] ——, "Aquila," https://model.baai.ac.cn/model-detail/100101, 2023.

[220] TigerResearch, "Tigerbot," https://github.com/TigerResearch/TigerBot/blob/main/README_en.md, 2023.

[221] OpenMEDLab, "Xraypulse," https://github.com/openmedlab/XrayPULSE, 2023.

[222] H. Xiong, S. Wang, Y. Zhu, Z. Zhao, Y. Liu, Q. Wang, and D. Shen, "Doctorglm: Fine-tuning your chinese doctor is not a herculean task," *arXiv preprint arXiv:2304.01097*, 2023.

[223] S. Diao, R. Pan, H. Dong, K. Shum, J. Zhang, W. Xiong, and T. Zhang, "Lmflow: An extensible toolkit for finetuning and inference of large foundation models," https://optimalscale.github.io/LMFlow/, 2023.

[224] P. Le and W. Zuidema, "Compositional distributional semantics with long short term memory," *arXiv preprint arXiv:1503.02510*, 2015.

[225] SenseTime, "Sensenova," https://www.sensetime.com/en/news-detail/51166397?categoryId=1072, 2023.

[226] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, *et al.*, "Constitutional ai: Harmlessness from ai feedback," *arXiv preprint arXiv:2212.08073*, 2022.

[227] S. Zhang, Q. Fang, Z. Zhang, Z. Ma, Y. Zhou, L. Huang, M. Bu, S. Gui, Y. Chen, X. Chen, and Y. Feng, "Bayling: Bridging cross-lingual alignment and instruction following through interactive translation for large language models," *arXiv preprint arXiv:2306.10968*, 2023. [Online]. Available: https://arxiv.org/abs/2306.10968

[228] S. Soni, M. Gudala, A. Pajouhi, and K. Roberts, "RadQA: A question answering dataset to improve comprehension of radiology reports," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference.*

Marseille, France: European Language Resources Association, June 2022, pp. 6250–6259. [Online]. Available: https://aclanthology.org/2022.lrec-1.672

[229] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[230] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[231] D. Demszky, K. Guu, and P. Liang, "Transforming question answering datasets into natural language inference datasets," *arXiv preprint arXiv:1809.02922*, 2018.

[232] A. S. White, P. Rastogi, K. Duh, and B. Van Durme, "Inference is everything: Recasting semantic resources into a unified evaluation framework," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 996–1005. [Online]. Available: https://aclanthology.org/I17-1100

[233] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[234] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[235] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford alpaca: An instruction-following llama model," https://github.com/tatsu-lab/stanford_alpaca, 2023.

[236] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[237] Z. Liu, A. Zhong, Y. Li, L. Yang, C. Ju, Z. Wu, C. Ma, P. Shu, C. Chen, S. Kim, *et al.*, "Radiology-gpt: A large language model for radiology," *arXiv preprint arXiv:2306.08666*, 2023.

[238] N. Muennighoff, T. Wang, L. Sutawika, A. Roberts, S. Biderman, T. L. Scao, M. S. Bari, S. Shen, Z.-X. Yong, H. Schoelkopf, X. Tang, D. Radev, A. F. Aji, K. Almubarak, S. Albanie, Z. Alyafeai, A. Webson, E. Raff, and C. Raffel, "Crosslingual generalization through multitask finetuning," 2022.

[239] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, *et al.*, "Bloom: A 176b-parameter open-access multilingual language model," *arXiv preprint arXiv:2211.05100*, 2022.

[240] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[241] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[242] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, "Publicly available clinical BERT embeddings," in *Proceedings of the 2nd Clinical Natural Language Processing Workshop.* Minneapolis,

Minnesota, USA: Association for Computational Linguistics, June 2019, pp. 72–78. [Online]. Available: https://www.aclweb.org/anthology/W19-1909

[243] OpenAI, “Gpt-4 technical report,” 2023.

[244] A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme, “Hypothesis only baselines in natural language inference,” *arXiv preprint arXiv:1805.01042*, 2018.

[245] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith, “Annotation artifacts in natural language inference data,” *arXiv preprint arXiv:1803.02324*, 2018.

[246] M. Tsuchiya, “Performance impact caused by hidden bias of training data for recognizing textual entailment,” *arXiv preprint arXiv:1804.08117*, 2018.

[247] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela, “Adversarial nli: A new benchmark for natural language understanding,” *arXiv preprint arXiv:1910.14599*, 2019.

[248] A. Stubbs and Ö. Uzuner, “Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus,” *Journal of biomedical informatics*, vol. 58, pp. S20–S29, 2015.

[249] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, “Is chatgpt a general-purpose natural language processing task solver?” *arXiv preprint arXiv:2302.06476*, 2023.

[250] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, *et al.*, “A comprehensive survey on pretrained foundation models: A history from bert to chatgpt,” *arXiv preprint arXiv:2302.09419*, 2023.

[251] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, *et al.*, “A multitask, multilingual, multimodal evalu-

ation of chatgpt on reasoning, hallucination, and interactivity," *arXiv preprint arXiv:2302.04023*, 2023.

[252] L. Reynolds and K. McDonell, "Prompt programming for large language models: Beyond the few-shot paradigm," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–7.

[253] E. Jiang, K. Olson, E. Toh, A. Molina, A. Donsbach, M. Terry, and C. J. Cai, "Promptmaker: Prompt-based prototyping with large language models," in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 2022, pp. 1–8.

[254] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, "Publicly available clinical bert embeddings," *arXiv preprint arXiv:1904.03323*, 2019.

[255] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how bert works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2021.

[256] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, no. 1, pp. 1–23, 2021.

[257] B. Pandey, D. K. Pandey, B. P. Mishra, and W. Rhmann, "A comprehensive survey of deep learning in the field of medical imaging and medical natural language processing: Challenges and research directions," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 5083–5099, 2022.

[258] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, *et al.*, “Opt: Open pre-trained transformer language models,” *arXiv preprint arXiv:2205.01068*, 2022.

[259] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, “Llm. int8 (): 8-bit matrix multiplication for transformers at scale,” *arXiv preprint arXiv:2208.07339*, 2022.

[260] M. Zhang and Y. He, “Accelerating training of transformer-based language models with progressive layer dropping,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 011–14 023, 2020.

[261] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, *et al.*, “Efficient large-scale language model training on gpu clusters using megatron-lm,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–15.

[262] Z. Zeng, C. Liu, Z. Tang, K. Li, and K. Li, “Acctfm: An effective intra-layer model parallelization strategy for training large-scale transformer-based models,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4326–4338, 2022.

[263] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu, “Biogpt: generative pre-trained transformer for biomedical text generation and mining,” *Briefings in Bioinformatics*, vol. 23, no. 6, 2022.

[264] S. Rezayi, Z. Liu, Z. Wu, C. Dhakal, B. Ge, C. Zhen, T. Liu, and S. Li, “Agribert: knowledge-infused agricultural language models for matching food and nutrition,” in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, vol. 7, 2022, pp. 5150–5156.

[265] W. Liao, Z. Liu, H. Dai, Z. Wu, Y. Zhang, X. Huang, Y. Chen, X. Jiang, D. Zhu, T. Liu, *et al.*, "Mask-guided bert for few shot text classification," *arXiv preprint arXiv:2302.10447*, 2023.

[266] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res*, vol. 23, pp. 1–40, 2021.

[267] S. Wang, Z. Zhao, X. Ouyang, Q. Wang, and D. Shen, "Chatcad: Interactive computer-aided diagnosis on medical image using large language models," *arXiv preprint arXiv:2302.07257*, 2023.

[268] S. Denève, A. Alemi, and R. Bourdoukan, "The brain as an efficient and robust adaptive learner," *Neuron*, vol. 94, no. 5, pp. 969–977, 2017.

[269] C. W. Lynn and D. S. Bassett, "How humans learn and represent networks," *Proceedings of the National Academy of Sciences*, vol. 117, no. 47, pp. 29 407–29 415, 2020.

[270] S. H. Tompson, A. E. Kahn, E. B. Falk, J. M. Vettel, and D. S. Bassett, "Individual differences in learning social and nonsocial network structures." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 45, no. 2, p. 253, 2019.

[271] S. Xie, A. Kirillov, R. Girshick, and K. He, "Exploring randomly wired neural networks for image recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1284–1293.

[272] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 261–24 272, 2021.

[273] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, “Metaformer is actually what you need for vision,” *arXiv preprint arXiv:2111.11418*, 2021.

[274] 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.

[275] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[276] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek, *et al.*, “Resmlp: Feedforward networks for image classification with data-efficient training,” *arXiv preprint arXiv:2105.03404*, 2021.

[277] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, “Training data-efficient image transformers &amp; distillation through attention,” in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 347–10 357. [Online]. Available: https://proceedings.mlr.press/v139/touvron21a.html

[278] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962.

[279] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[280] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Citeseer*, pp. 32–33, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[281] H. Song, M. Kim, and J.-G. Lee, "SELFIE: Refurbishing unclean samples for robust deep learning," in *ICML*, 2019, pp. 5907–5915.

[282] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1447–1454.

[283] R. Wightman, "Pytorch image models," https://github.com/rwightman/pytorch-image-models, 2019.

[284] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," *arXiv preprint arXiv:2104.10972*, 2021.

[285] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. International Society for Optics and Photonics, 2019, p. 1100612.

[286] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[287] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.

[288] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[289] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.

[290] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[291] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[292] A. Tealab, "Time series forecasting using artificial neural networks methodologies: A systematic review," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 334–340, 2018.

[293] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 819–10 829.

[294] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[295] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, "A comprehensive survey of neural architecture search: Challenges and solutions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–34, 2021.

[296] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019.

[297] Y. Zhang, M. Choi, K. Han, and Z. Liu, "Explainable semantic space by grounding language to vision with cross-modal contrastive learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 513–18 526, 2021.

[298] D. S. Bassett, N. F. Wymbs, M. P. Rombach, M. A. Porter, P. J. Mucha, and S. T. Grafton, “Task-based core-periphery organization of human brain dynamics,” *PLoS computational biology*, vol. 9, no. 9, p. e1003171, 2013.

[299] S. Gu, C. H. Xia, R. Ciric, T. M. Moore, R. C. Gur, R. E. Gur, T. D. Satterthwaite, and D. S. Bassett, “Unifying the notions of modularity and core–periphery structure in functional brain networks during youth,” *Cerebral Cortex*, vol. 30, no. 3, pp. 1087–1102, 2020.

[300] R. J. Gallagher, J.-G. Young, and B. F. Welles, “A clarified typology of core-periphery structure in networks,” *Science advances*, vol. 7, no. 12, p. eabc9800, 2021.

[301] P. Barberá, N. Wang, R. Bonneau, J. T. Jost, J. Nagler, J. Tucker, and S. González-Bailón, “The critical periphery in the growth of social protests,” *PloS one*, vol. 10, no. 11, p. e0143611, 2015.

[302] J. I. Alvarez-Hamelin, L. Dall’Asta, A. Barrat, and A. Vespignani, “K-core decomposition of internet graphs: hierarchies, self-similarity and measurement biases,” *arXiv preprint cs/0511007*, 2005.

[303] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir, “A model of internet topology using k-shell decomposition,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 27, pp. 11 150–11 154, 2007.

[304] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, “Identification of influential spreaders in complex networks,” *Nature physics*, vol. 6, no. 11, pp. 888–893, 2010.

[305] S. P. Borgatti and M. G. Everett, “Models of core/periphery structures,” *Social networks*, vol. 21, no. 4, pp. 375–395, 2000.

[306] Tang, Yehui and Han, Kai and Wang, Yunhe and Xu, Chang and Guo, Jianyuan and Xu, Chao and Tao, Dacheng, “Patch slimming for efficient vision trans-

formers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 165–12 174.

[307] Y. Xu, Z. Zhang, M. Zhang, K. Sheng, K. Li, W. Dong, L. Zhang, C. Xu, and X. Sun, "Evo-vit: Slow-fast token evolution for dynamic vision transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 2964–2972.

[308] M. Zhu, Y. Tang, and K. Han, "Vision transformer pruning," *arXiv preprint arXiv:2104.08500*, 2021.

[309] H. Yu and J. Wu, "A unified pruning framework for vision transformers," *arXiv preprint arXiv:2111.15127*, 2021.

[310] T. Chen, Y. Cheng, Z. Gan, L. Yuan, L. Zhang, and Z. Wang, "Chasing sparsity in vision transformers: An end-to-end exploration," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19 974–19 988, 2021.

[311] M. Chen, H. Peng, J. Fu, and H. Ling, "Autoformer: Searching transformers for visual recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 270–12 280.

[312] X. Su, S. You, J. Xie, M. Zheng, F. Wang, C. Qian, C. Zhang, X. Wang, and C. Xu, "Vitas: Vision transformer architecture search," *arXiv preprint arXiv:2106.13700*, 2021.

[313] P. Erdos, A. Rényi, *et al.*, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

[314] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[315] R. Walker, "Implementing discrete mathematics: combinatorics and graph theory with mathematica, steven skiena. pp 334. 1990. isbn 0-201-50943-1 (addison-wesley)," *The Mathematical Gazette*, vol. 76, no. 476, pp. 286–288, 1992.

[316] 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.

[317] M. Cucuringu, P. Rombach, S. H. Lee, and M. A. Porter, "Detection of core–periphery structure in networks using spectral methods and geodesic paths," *European Journal of Applied Mathematics*, vol. 27, no. 6, pp. 846–887, 2016.

[318] P. Holme, "Core-periphery organization of complex networks," *Physical Review E*, vol. 72, no. 4, p. 046111, 2005.

[319] M. P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha, "Core-periphery structure in networks," *SIAM Journal on Applied mathematics*, vol. 74, no. 1, pp. 167–190, 2014.

[320] H. Liu, S. Zhang, X. Jiang, T. Zhang, H. Huang, F. Ge, L. Zhao, X. Li, X. Hu, J. Han, *et al.*, "The cerebral cortex is bisectionally segregated into two fundamentally different functional units of gyri and sulci," *Cerebral Cortex*, vol. 29, no. 10, pp. 4238–4252, 2019.

[321] X. Chen, C.-J. Hsieh, and B. Gong, "When vision transformers outperform resnets without pre-training or strong data augmentations," *arXiv preprint arXiv:2106.01548*, 2021.

[322] I. C. Moreira, I. Amaral, I. Domingues, A. Cardoso, M. J. Cardoso, and J. S. Cardoso, "Inbreast: toward a full-field digital mammographic database," *Academic radiology*, vol. 19, no. 2, pp. 236–248, 2012.

[323] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[324] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[325] S. Wang, X. Ouyang, T. Liu, Q. Wang, and D. Shen, "Follow my eye: Using gaze to supervise computer-aided diagnosis," *IEEE Transactions on Medical Imaging*, 2022.

[326] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[327] Y. Wang, R. Huang, S. Song, Z. Huang, and G. Huang, "Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 960–11 973, 2021.

[328] B. Ibrokhimov and J.-Y. Kang, "Two-stage deep learning method for breast cancer detection using high-resolution mammogram images," *Applied Sciences*, vol. 12, no. 9, p. 4616, 2022.

BIOGRAPHICAL STATEMENT

Lu Zhang received her Ph.D. in Computer Science and Engineering from the University of Texas at Arlington at 2024. Prior to beginning the Ph.D. program, Lu received her bachelor's and master's degrees in computer science and technology from the Northwestern Polytechnical University in China. Her research topic mainly focuses on two major problems: using cutting-edge AI techniques to explore brain fundamental organization principles and investigate brain disorders, such as Alzheimer's disease; and leveraging superior organizational principles in brain networks to inspire and design more effective and efficient deep neural networks (Brain-Inspired-AI). Currently, Lu has published more than thirty papers on top-tier conferences and Journals in the brain imaging area, including Medical Image Analysis, Cerebral Cortex, TNNLS, MICCAI, ISBI, and Alzheimer's Dementia. In addition, Lu has received the Best Paper Award at MMMI 2019, the UTA Doctoral Student Research Travel Grant Award in 2019, Student Travel Award at MICCAI 2020, the Prestigious Young Scientist Award at MICCAI 2020, and the Outstanding Doctoral Dissertation Award in 2024.