Imperial College London

MSc AI Individual Project

Imperial College London

Department of Computing

Graph Neural Network GAN: Data Augmentation for Alzheimer's disease classification tasks

Author:
Yikang Li

Supervisor:
Dr. Chao Li
Dr. Stamatia Giannarou

Second Marker:
Dr. Bjoern Schuller

August 31, 2022

Abstract

As one of the most common age-related dementia diseases, Alzheimer's disease (AD) significantly affects patients' daily life and impact socio-economics. Brain diffusion MRI has been proved as an essential biomarker to detect brain status, further indicating the individuals at risk of dementia. Recent researchers have applied the deep learning approach to classification tasks, showing the superiority in shortening the time and enhancing the precision. However, most of the previous research mainly considers the end-to-end convolution neural network(CNN) as the primary classification architecture, which ignores the structural and topological features of the Brain MRI. Moreover, the limited availability of diffusion MRI with unbalanced labels challenges the model training of deep learning. In this case, we wish to utilize the Graph Neural Network to improve the performance of the Alzheimer's disease classification task since the GNN is intuitively beneficial in handling graph datasets. Nevertheless, we also substitute the classical CNN discriminator and classifier of the GAN with the GNN discriminator and classifier to efficiently augment the structural brain networks for dementia classifying tasks.

Goals: We set out to (1) connect the novel graph machine learning approaches with the existing basic generative adversarial network architecture. (2)Pursue un-trialed graph machine learning previous approaches for effectively capturing the topological features of the brain connectivity matrices. (3)Propose, implement and evaluate the superiority of the various novel combinations of the relevant elements. As the main component of the GAN, we implement a novel architecture that combines both superiority of GNNs and node spatiality. In that case, it not only improves the accuracy of the Alzheimer's Disease(AD) classification task but also proves the availability of adding graph neural network approaches to the existing GAN. To perform the quantitative analysis, our method was tested on three different Brain MRI datasets collected for binary dementia classification tasks.

Key Results: Our novel spatial node feature with GNNs has reached state-of-the-art performance compared to the other baseline CNN and GNNs for the Alzheimer's Disease classification tasks. This model applied a novel graph spatiality computation method that combines spatial and topological information of diffusion Brain MRI.

Acknowledgements

To my beloved grandfather, thank you so much for everything you have done for our family. I pray that you can get better soon.

Thanks to my parents, who never abandoned me and trusted me all the time. I could not accomplish the master's degree without your selfless support.

Thanks to all my supervisors that supported my work. There is no way to accomplish this work without your suggestions.

I also wanna say thanks to myself that I never gave up on myself, even in the darkest period.

Contents

Chapter 1

Introduction

1.1 Objectives

As one of the most common age-related dementia, Alzheimer's disease is lethal for patients' cognitive and socioeconomic performance. Around 2% - 3% population in industrialized countries has been affected by Alzheimer's disease.[Ala18] Usually, people over 70 years old have a higher risk of Alzheimer's disease. However, the patients with AD are becoming younger and more universal in the present. Moreover, the incidence of AD will increase threefold within the next 50 years, according to the predictions of scholars. The main symptom of Alzheimer's disease is the loss of memory caused by the degeneration of synapses and the death of neurons. The human brain contains different regions which are characterized by their function. In general, the learning and memory processes of brain regions such as the temporal and frontal lobes of AD patients are reduced in pathological. In this case, it is essential to analyze the Brain Magnetic Resonance Imaging(MRI) to classify whether an individual is a patient. Brain MRI can be divided into two parts, diffusion MRI (dMRI or DTI) and functional MRI (fMRI). Usually, we use diffusion MRI to diagnose the status of patients' brains since the dMRI allows tissue structure to be probed and imaged on a microscopic scale which provides some unique clues to the fine architecture of neural tissues and changes associated with diverse physiological and pathological conditions.

In parallel, machine learning approaches for dementia classification tasks to distinguish AD patients from healthy people have shown their superiority. Mainly, state-of-the-art deep learning models have been widely used in predicting dementia using end-to-end training schemes such as CNN. The training dataset for the dementia classification task is usually the brain connectivity matrix generated from the Brain diffusion MRI. The connectivity matrix can be considered as a homogeneous graph when we divide the diffusion MRI into atlas regions. Otherwise, the connectivity matrix can also be considered as a heterogeneous graph once we generate the connectivity matrix based on the brain regions of fMRI. Usually, the brain connectivity matrix is demonstrated in a graph form in which nodes are represented the brain atlas, and the edges represent the status of connections between brain atlas regions. Then it is natural to utilize the graph neural network for Alzheimer's Disease classification task. Besides that, it is necessary to require a large amount of training data with balanced class labels for good classification performance for any deep learning models. Then graph adversarial networks are needed for brain data augmentation.

1.2 Challenges

The main challenges are fourfold.

1. As one of the most common diseases, Alzheimer's disease (AD) has a significant impact on patients' cognitive performance and socioeconomic status. For curing this disease, numerous classification tasks are essential.

2. Although the deep learning approach has been proven useful for finishing the classification tasks, the limited Alzheimer's disease(AD) datasets also restricted the development of deep learning methods.

3. Although the Synthetic Minority Oversampling Technique(SMOTE) and Adaptive Synthetic

Sampling (ADASYN) are also widely used, their KNN-assisted techniques have been proved less effective for capturing the topological features.

4. We have proved that standard generative adversarial networks can also be used to finish the data augmentation tasks. However, the classical CNN discriminator and classifier are not suitable for handling the irregular Brain network data

1.3 Contributions

1. We propose a novel model architecture that combines both node spatiality and graph neural network, which reaches state-of-the-art performance.

2. We have compared our method with several existing classification methods, including baseline CNN and GNN, that clearly demonstrated the development we have made.

3. We have also completed a comparative analysis of several existing classification GNNs and many hand-designed node features. In this case, we have enhanced the AI interpretability of our novel architecture. We also experiment with multiple combinations of the related approaches.

4. Our combined approach also improves the performance of the data augmentation task by effectively capturing the structural and topological features of Brain Network data.

1.4 Outline of Report

The report starts by listing and analyzing the relevant past and latest approaches related to classification GNNs and brain network data augmentation task(2). Secondly, we explored the data used in this project, including the data characteristics and pre-processing steps(3). Thirdly, We have demonstrated and discussed the various metrics(4) and methods(5) used in the evaluation and discussion parts. Fourth, we have shown the quantitative and qualitative results of several related Graph Neural Network approaches. Finally, we have made our final conclusions and summaries based on the results and analysis(7).

1.5 Ethical Consideration

Since the immediate context of this project regards Alzheimer's Disease brain networks from diffusion MRI scans of patients, which contain sensitive personal information, it is necessary to introduce multiple ethical considerations for protecting the biological and personal information of patients. The most important consideration is to anonymize all MRI data to avoiding identify the identities of the patients by reverse engineering. Moreover, all data utilized in this project are under the patient's consent. It is prohibited to use the data without permission from the patients.

Otherwise, all the data was fully processed to anonymise the personal information. The raw data of brain networks is stored in the DICOM version, which contains identifiable header tags. However, that information is not helpful for the project. In this case, all data were transformed into different formats and removed that information. In our case, all diffusion MRI data were transformed into brain connectivity matrix form, which does not contain personal information.

As for the store issue, most of the datasets we have utilized in this project are public. Moreover, all these datasets were fully anonymised and under patients' consent. In these cases, storage of the datasets will not be a severe problem for ethical consideration.

Chapter 2

Background

In this Background chapter, most of the relevant past and current research related to the topic of the classification of Graph Neural Networks is presented and explained clearly. The major challenge of the graph classification task is identifying whether the connectivity matrix is Alzheimer's disease or ordinary people based on the objects of interest - in our case, the pathological regions (nodes and edges). In this chapter, Alzheimer's Disease, mainly used toolboxes, classical approaches, and the latest research approaches are all briefly explored, providing the context for the discussion of Graph machine learning and spatial embedding methods.

2.1 Alzheimer's Disease

As one of the most common dementia diseases globally, Alzheimer's disease is usually considered a progressive neurodegenerative disease.[EXXZ19] The main symptoms of Alzheimer's disease are usually associated with memory deficits and cognitive decline, which finally leads to a lethal impact on the patient's daily life and socioeconomics. Unfortunately, there is no conclusive evidence of the primary causes of the diseases except in a small number of familial cases, such as genetic mutations. In these cases, it remains no effective treatment options for Alzheimer's disease patients. In discovering the treatment of Alzheimer's disease, people have realized that Alzheimer's disease is a mixed proteinopathy (amyloid and tau) usually associated with other age-related processes such as cerebrovascular disease. Usually, people over 70 years old have a higher risk of Alzheimer's disease. However, the patients with AD are becoming younger and more universal in the present. Moreover, the incidence of AD will increase threefold within the next 50 years, according to the predictions of scholars.[Cit10]

The main physiological feature of Alzheimer's disease brains is the shrinking of the brain and the death of brain neurons. In particular, the brain regions that control learning and memory, including the temporal and frontal lobes, are mostly affected.[LK21] However, as mentioned above, since most of the patients are over 70 years old, there are no convincing causes of memory loss that can be demonstrated before the postmortem examination of the brain. Sufficient numbers of "plaques" and "tangles" are required for the examination, which can qualify whether the patients are affected by Alzheimer's disease. Most of the "plaques" and "tangles" are present mainly in learning, memory, and emotional brain regions. In an anatomical view, the entorhinal cortex, hippocampus, basal forebrain, and amygdala are the most affected brain regions.[MNG+22]

Unfortunately, Alzheimer's disease is one of the largest unsolved medical problems in neurology. Although some of the current drugs or other treatment methods are able to relieve the symptoms, there still does not exist an instant solution for eradicating the disease. In recent years, some of the medical approaches aimed at inhibiting Alzheimer's disease progression have advanced to clinical trials, which accomplished an exhilarating step. However, most medical approaches required a huge amount of human involvement, which made these methods expensive and time-consuming.[Wen06] In these cases, AI-involved treatments are mainly considered and developed recently. For the AI-involved treatment, the key points mainly contain the training dataset and learning model.[WLP21] For the training dataset, medical imaging methods, namely magnetic resonance imaging and, in particular, diffusion MRI, have proved their efficiency and reliability by integrating several rapidly developing fields of science and engineering. Nevertheless, numerous studies have also reported the usefulness and reliability of the brain connectivity matrix in characterising and classifying neuro-

logical diseases, especially Alzheimer's disease. In detail, the brain connectivity matrix constructed from the diffusion MRI brings a more accurate brain imaging that clearly states the topological structure of the brain, contributing to narrowing the gap between human understanding of the brain and human behavior or cognition.[SMR+08] In addition, MRI is mainly divided into two types diffusion MRI and functional MRI. In this work, we mainly considered the dMRI as our training dataset, and we also set the fMRI as a reference to test the reliability and robustness of the model.[Ala18] In this work, we utilized the dMRI to quantify the brain white matter structure and incorporate the prior knowledge of brain anatomy. Moreover, the tractography performed on dMRI was utilized to quantify the connectivity strength between the separated anatomical brain atlas. [DD19]

Recent studies have shown the high efficiency of the machine learning approaches based on the structural network in classification tasks on distinguishing Alzheimer's disease patients from healthy controls. In particular, the end-to-end training technique has reached state-of-the-art performance on dementia prediction tasks.[GBTH13] However, most previous works were convolution neural networks that cannot effectively capture the topological features of the brain network, which leads to a reduction in the final performance. In this case, graph neural network (GNN) architecture for brain classification was proposed in the latest period. Otherwise, the data augmentation task is also essential for the classification task since deep learning models usually require a tremendous amount of training data with balance labels. However, the traditional data augmentation methods, such as rotation, flipping, or crop, cannot be applied to the data network datasets. Thus, the classification task and data augmentation task are both desired for our project.[LWC21]

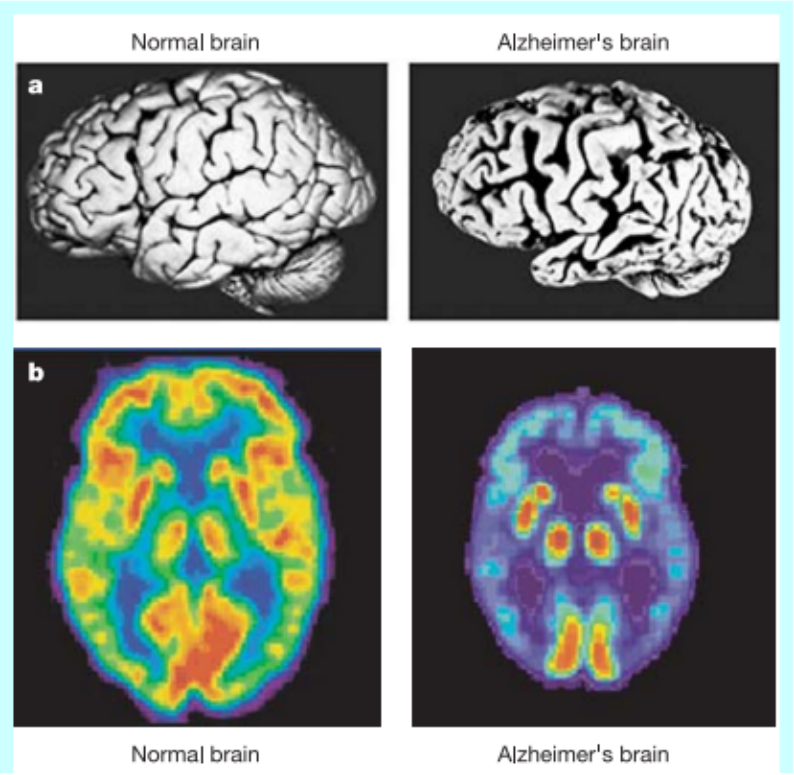

Figure 2.1: Visualize AD and CN [Ala18]

2.2 BrainNet CNN

The BrainNetCNN is a convolutional neural network(CNN) framework instead of a graph neural network to predict clinical pathological outcomes of brain networks. It consists of novel edge-to-edge, edge-to-node, and node-to-graph convolutional filters that can effectively capture and leverage the topological locality of structural brain networks. [KBM+16]

As shown in 2.2, each block in the architecture represents the input and output of the numbered filter layers. The third dimension of each block demonstrates the number of feature maps, M, at that stage. The input brain network adjacency matrix is first convolved with the novel E2E(edge to edge) filters which weigh the edges of adjacent brain regions. The response is convolved with an E2N(edge to node) filter, which assigns each brain region a weighted sum of its edges. The N2G(node to graph) assigns a single response based on all the weighted nodes. Finally, fully connected (FC) layers generate predictions by reducing the number of features.

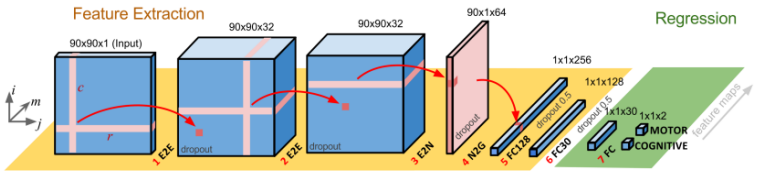

Figure 2.2: Brainnet CNN[KBM+16]

2.3 Graph Neural Network

In this project, the significant development we made was based on the graph neural network. The first reason we consider GNN as our primary research object is that the brain connectivity matrix is usually considered a graph form, which intuitively means GNN can perform better on both the classification and augmentation tasks.

For any data structure, the data can be expressed as a graph, $G = (V, E, A)$ which contains a set of nodes $i \in V$, a set of edges $e_{ij} \in E$ and adjacency matrix A. Nodes V represent the graph's vertices and edges E represent the connections between the vertices. Usually, if $e_{ij} = 1$, it means there is a connection between nodes i and j. If $e_{ij} = 0$, it means there is no connection between nodes i and j. This connection information can be summarized as the corresponding adjacency matrix, A. [SGT+09] [SLRPW21] In our case, the nodes i and j in the brain network can represent as brain atlas region. The edge e_{ij} indicates the connections between different brain atlas regions.

For the classification task, given a set of graphs $\{G_1, ..., G_N\} \subseteq \mathcal{G}$ and labels $\{y_1, ..., y_N\} \subseteq \mathcal{Y}$, the aim is learning a representation vector h_G that helps to predict the entire graph, $y_G = g(h_G)$. For the classical GNN, the mathematics formula can be represented as follows:

$a^(k)_v = aggregate^(k)(\{h^{k-1})_u : u \in \mathcal{N}_v\})$, $h^k_v = combine^(k)(h^(k-1)_v, a^(k)_v)$ Where h^k_v is the feature vector of node v in k-th layer, it can iteratively update the node feature by aggregating all the nodes' neighborhood information.

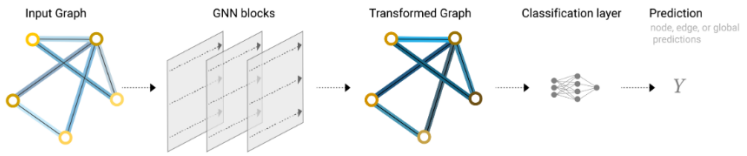

Figure 2.3: GNN for classification Task [SLRPW21]

2.3.1 Graph Convolution Network

The graph convolution network follows the similar architecture of the GNN except for the graph convolution layers. Moreover, the element-wise mean pooling is also applied to aggregate and combine the node representations.[KW16] The mathematical formula of GCN can be expressed as follows: $h^k_v = ReLU(W^k Mean\{h^(_u k-1)|u \in \mathcal{N}_v \bigcup v\})$ which h^k_v is the node representation of the final iteration for the graph classification task.

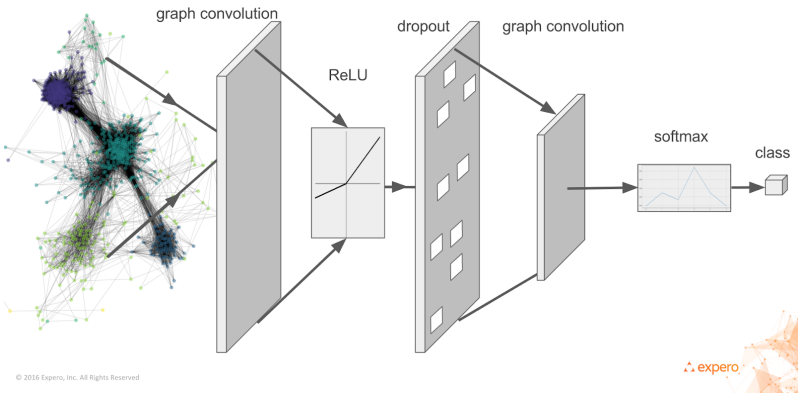

Figure 2.4: GCN for classification Task [Gan18]

2.3.2 Graph Isomorphism Network

For exploring the maximum representational capacity of GNN, the graph isomorphism network(GIN) was implemented that follows the Weisfeiler-Lehman (WL) graph isomorphism test.[?] The main idea of GIN is mapping the isomorphic graphs to the exact representation and non-isomorphic graphs to another different representation. For proving the statement, a theorem about the WL test has been proposed. The theorem states that Let G_1 and G_2 be any two $non-isomorphism$ graphs. If a graph neural network $\mathcal{A} : \mathcal{G} \Rightarrow \mathcal{R}^d$ maps $\mathcal{G}_1$ and $\mathcal{G}_2$ to different embeddings, the Weisfeiler-Lehman graph isomorphism test also decides that G_1 and G_2 are not $isomorphic$. It states that if the neighbor aggregation and graph-level readout functions are injective, then the resulting GNN is as powerful as the WL test.[XHLJ18] Moreover, GIN also achieves maximum discriminative power among GNNs by generalizing the WL test. Thus, GIN can be interpreted as follows: $h_v^k = MLP^k((1 + \epsilon^k * h_v^k - 1) + \sum_{\mu \in \mathcal{N}_v} h_u^{k-1})$, which h_v^k is the final node representation learned by multi-layer perceptrons.

2.3.3 Graclus

Graclus algorithm was mainly implemented for graph clustering problems since it can effectively optimise various weighted graph clustering objectives, such as the ratio cut and normalized cut. The main idea of Graclus is that it identified different pairs of maximally similar nodes and clustered them together to form a new node or vertex. Then the max pooling is also utilized for determining the node attribute at each iteration of the GNN. Then the generated nodes are added to make sure the number of nodes is halved each time. Nevertheless, Graclus can also be utilized for graph classification tasks if we treat it as a pooling operation in GNNs.[DGK07] [BGA20]

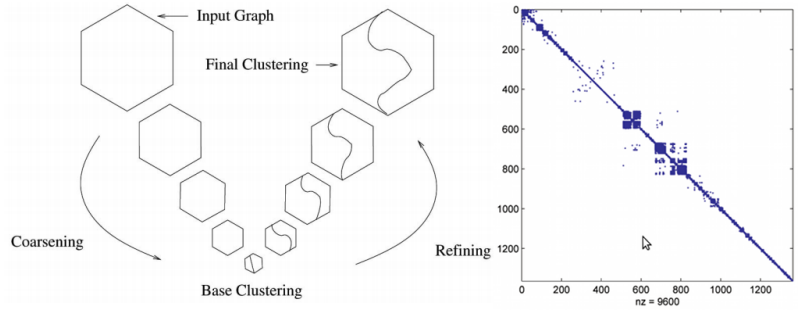

Figure 2.5: Graclus Algorithm [DGK05]

2.3.4 TopK

Suppose there are N nodes in graph G that contains M features. Then the graph can be represented as adjacency matrix $A^{\updownarrow} \in \mathcal{R}^{N \times N}$ and feature matrix $X^{\updownarrow}\ in \mathcal{R}^{N \times N}$. Then the top-K pooling layer can be stated as follows: $y = \frac{X_p}{||p||}$, $i = top_k(y)$, $X' = (X \odot sigmoid(y))_i$, $A' = A_{i,i}$. It states that $\odot$ denotes element-wise matrix multiplication, and $\mathbf{p}$ is a learnable projection vector. The projection vector $\mathbf{p}$ computes a scalar projection value for each graph node. The nodes with the top-k highest projection scores are retained in the new adjacency matrix $\mathbf{A}'$.The sigmoid$(\cdot)$ operation makes the projection vector $\mathbf{p}$ trainable by backpropagation.[GJ21]

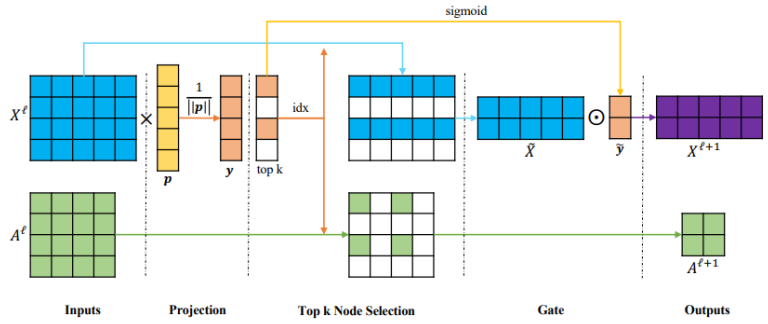

Figure 2.6: Top-K[GJ21]

2.3.5 Self-Attention Graph Pooling

After the self-attention mechanism has been proposed, it makes it possible that the model can focus more on the essential features and less on the minor features. [VSP+17] It was implemented from the basic GNN and top-K pooling algorithm, but it can compute both graph features and graph topology. The Self-Attention Graph Pooling algorithm can be interpreted as follows: $y = GNN(X, A)$, $i = top_k(y)$, $X' = (X \odot y)_i$, $A' = A_{i,i}$ which $A^{\updownarrow} \in \mathcal{R}^{N \times N}$ represents the adjacency matrix, and $X^{\updownarrow}\ in \mathcal{R}^{N \times N}$ represents the feature matrix for graph G with N nodes and M features. $\odot$ denotes element-wise matrix multiplication, and GNN is a generic permutation equivariant GNN layer. [LLK19]

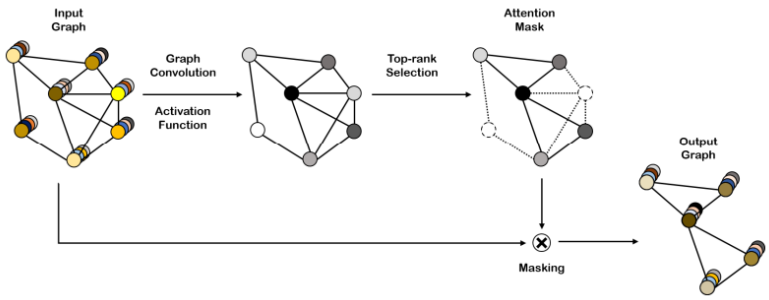

Figure 2.7: Self-Attention Pooling[LLK19]

2.3.6 Edge Pooling

Edge Pooling combines edge contraction, an important transformation operation in the graph theory field, with the end-to-end learning mechanism. In that case, it achieved the hierarchical pooling operation on the graph data. In summary, it iteratively merges the nodes on each pairwise edge to form a new node while preserving the connection relationship between the first two nodes and the new node. Then, Edge Polling must design a score for each edge and performs non-repetitive selection and merging based on the score. The operation can be represented as follows:

$r_{ij} = w^T[h_i||h_j] + b$ for calculating the initial score of each edge. Then, $s_{ij} = softmax_j(r_{ij})$ states the normalization of the initial score along the neighborhood nodes. Then it sorts all s_{ij} for picking the two unselected nodes with the highest score in turn to perform the contraction. Finally, it aggregate the node features in this way, $h_{ij} = s(h_i, h_j)$ which $s = max(s_{ij}, s_{ji})$. [Die19]

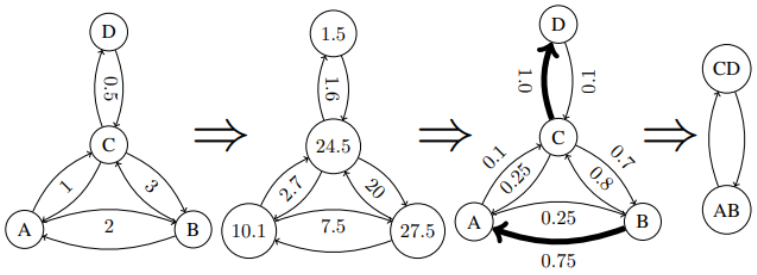

Figure 2.8: Edge Pooling[Die19]

2.3.7 Set2Set Network

The set2set model is mainly implemented to operate the sets dataset, and it is more powerful compared to the simple summation of the final node states. It firstly applies a linear projection to each tuple (h_v^T, x_v) and then takes as input the set of projected tuples $T = (h_v^T, x_v)$. Then the set2set model generates a graph level embedding q_t^*, which is invariant to the order of the tuple T after M steps of computation. Finally, the output is produced by feeding the q_t^* through a neural network. [VBK16] [GSR+17]

2.3.8 Sort Pooling

The Sort Pooling architecture is mainly implemented to sort the feature descriptors, which represent each vertex. For graph datasets, the vertices or nodes can be ordered according to their feature in the graph structure. The Weisfeiler-Lehman algorithm uses vertex colors by way of preprocessing to define the order of vertex-based graph topology so that vertices with the same color have a similar topology.

As a bridge between the graph convolution layer and the traditional layer, SortPooling also has a great advantage that it can return the loss gradient to the previous layer by memorizing the sorting order of the input. In that case, the training of the previous layer parameters is feasible. In contrast, since the WL algorithm sorts vertices in the preprocessing step, parameter training cannot be done before sorting.[ZCNC18]

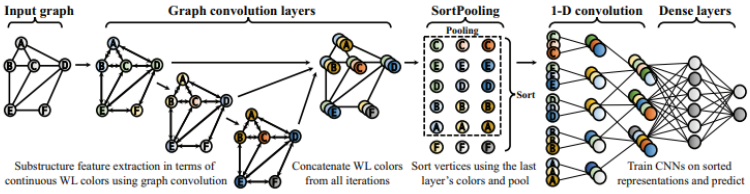

Figure 2.9: Sort Pooling[ZCNC18]

2.3.9 Adaptive Structure Aware Pooling

Adaptive structure-aware pooling(ASAP) is a sparse and differentiable pooling method that utilizes the self-attention mechanism based on the modified GNN formulation. It does not only address the limitations of previous graph polling architectures but also effectively captures the importance of each node in a graph. Moreover, it also learns a sparse soft cluster assignment for nodes at each layer to effectively pool the subgraphs to form the pooled graph.

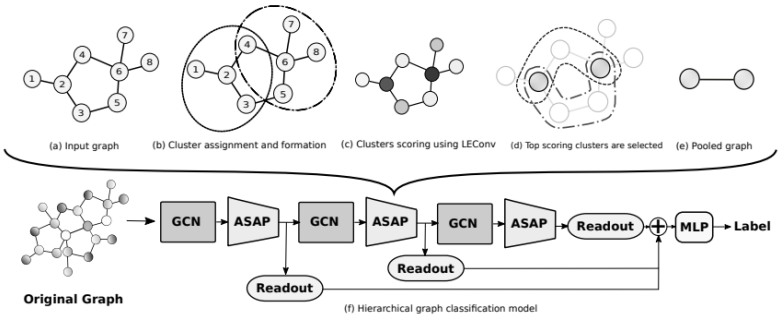

Figure 2.10: Adaptive Structure Aware Pooling[RST20]

As shown in the figure 2.10, it initially clusters a 1-hop neighborhood considering all nodes as medoids once input a graph. Then assign the scores to all clusters using LEConv. A darker shade means a higher score. After that, a fraction of the top scoring clusters is selected in the pooled graph. The edge weights between the member nodes inside the selected clusters will be utilized to recompute the adjacency matrix. Finally, it generates a pooled graph. [RST20]

2.3.10 BrainGNN

The BrainGNN is mainly implemented to analyze functional magnetic resonance images(fMRI), especially brain images. For discovering the neurological biomarkers of the brain graphs, it proposed a novel ROI-aware graph convolutional(Ra-GConv) layer that can effectively capture the topological and functional information, as shown in 2.13. Moreover, it also proposed a novel ROI-selection pooling layer(R-Pool) that both ensures the transparency of medical image analysis and highlights the meaningful salient ROIs, as shown in 2.14. [LZD+21]

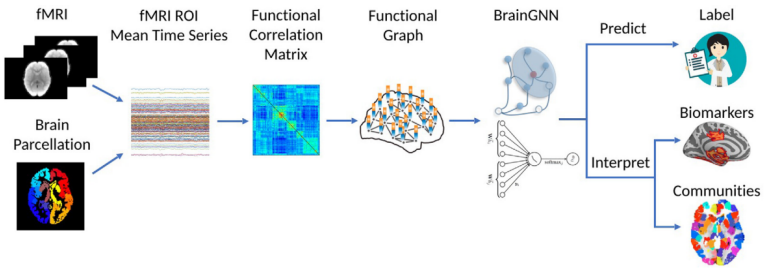

Figure 2.11: BrainGNN Pipeline[LZD+21]

As shown in 2.11, the fMRI images are decomposed by the atlas and transferred to the graph. Then, the graph is sent to BrainGNN, which gives task-specific predictions. BrainGNN collectively selects important brain regions useful for prediction tasks and clusters brain regions into prediction-related functional areas.

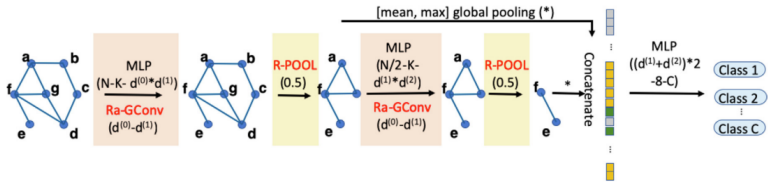

Figure 2.12: BrainGNN Architecture[LZD+21]

As shown in 2.12, it demonstrates the central architecture of BrainGNN. BrainGNN consists of Ra-GConv and R-pool blocks. It takes a graph as input and outputs graph-level predictions.

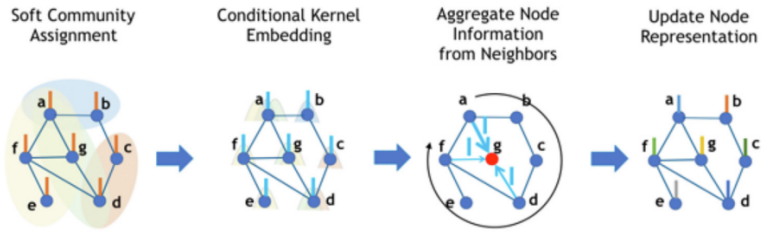

Figure 2.13: Ra-GConv Layer[LZD+21]

The Ra-GConv layer assigns all nodes to their belonging communities based on their membership scores. It can be regarded as node clustering. Then each node is embedded by a particular basis vector which is associated with each community. Finally, it assigns the updated representation to each node after aggregating a node's embedding and its neighbors' embedding, as shown in 2.13.

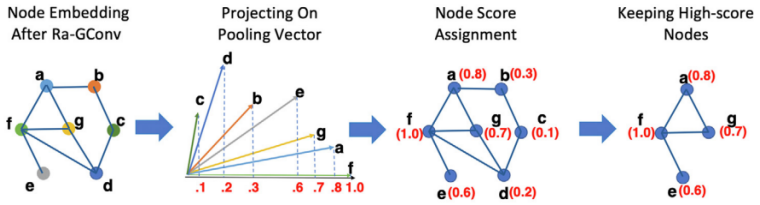

Figure 2.14: R-Pool Layer[LZD+21]

The R-Pool layer firstly projects all the nodes' representations to a learnable vector. Then the nodes are retained with their corresponding connections if they have large project values. In this case, the substantial salient ROIs(nodes) can be highlighted, as shown in 2.14.

2.4 Spatial Graph Convolution Network

Although Graph Convolutional networks (GCN) are now widely used in many fields, it remains the problem of lacking the ability to take into account the ordering of node neighbors. In that case, Spatial Graph Convolutional Network(SGCN) is mainly implemented for capturing the spatial features to efficiently learn from graphs that can be naturally located in space. Thus, the SGCN can leverage the node position to consider the spatiality of graphs. Moreover, the SGCN also improves the performance of graph classification tasks. [DST+20]

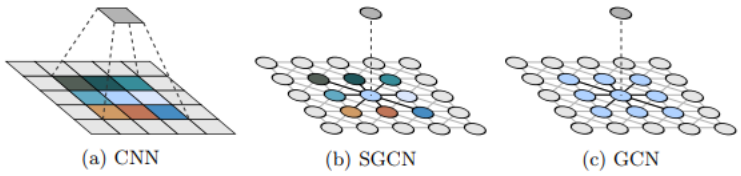

Figure 2.15: Spatial Graph Convolutional Network[DST+20]

2.5 Deep Generative Models

For the deep generative models, the most important one is the generative adversarial network (GAN). It is implemented by Ian Goodfellow and his colleagues. Now it is widely used in many

fields with numerous different variants of GAN. The main idea of GAN is from the game theory, which requires two agents. In this case, the architecture of GAN consists of two neural networks that represent the generator and discriminator. The generator outputs the generated data based on the real dataset, and the discriminator output a probability of whether the input data is real or fake.

Mathematically, the formula of GAN can be interpreted as follows:

$L(\mu_G, \mu_D) := E_{x\sim\mu_{ref}, y\sim\mu_D(x)}[\ln y] + E_{x\sim\mu_G, y\sim\mu_D(x)}[\ln(1-y)]$,

which μ_G represents the generator strategy set and μ_D represents the discriminator strategy set. For achieving the zero-sum game, the generator aims to minimize the objective function, that is, minimizing the distance between the generated data and the real data. The discriminator aims to maximize the objective function, that is, maximizing the distance between the generated data and real data. [GPAM+14]

The main reasons that we are utilizing the GAN for accomplishing the brain data augmentation task are folds. The first one is that the classical data augmentation methods, such as flipping or cropping, are suitable for brain network data because their structure and topology features are essential for the classification task. The second one is that the generated data does look not only similar to the real data but also has multiple classes compared to the real data.

2.6 Geometric Deep Learning

The main toolbox we use is the PyTorch geometric learning package. It is an open-source library built upon PyTorch to easily write and train Graph Neural Networks (GNNs) for a wide range of applications related to structured data. It consists of various methods for deep learning on graphs and other irregular structures, also known as deep geometric learning, from a variety of published papers. In addition, it consists of many mini-batch loaders for operating on many small and single giant graphs. Moreover, it also contains some example datasets for testing the generalization of the GNN models. [FL19]

2.7 Summary

In conclusion, most of the graph classification methods that relate to the brain network have been explored clearly in this section. From the previous approaches explored, there is one main innovation that appears:

1. Graph machine learning: Since the brain connectivity matrix is more suitable to be treated as graph data, it is intuitive to apply different GNN-based networks to improve the disease classification task and the data augmentation task. The previous CNN-based approaches, such as the BrainNet CNN, come with the disadvantage of ignoring the topological features of the brain network data. More specifically, most CNN-based models treat the brain network data as 2-D images, which are clearly not fittable. In contrast, the GNN-based approaches can effectively capture the structural and topological features of the brain network data. Thus, various GNN-based methods will be explored and applied, with novel approaches being compared to existing techniques.

Chapter 3

Data Sources and Pre-processing

In this chapter, I will explore the datasets I have utilized in this project. Although the primary dataset is the Alzheimer's Disease(AD) dataset, I still applied two additional datasets, the Alzheimer's Disease Neuroimaging Initiative(ADNI) dataset and the resting-state fMRI (rsfMRI) dataset, to validate the robustness and liability of the GNN-based methods. Moreover, the pre-processing steps are also described and later discussed in this chapter.

3.1 Data

The focus of this research is Graph machine learning applied to the task of Alzheimer's disease classification and data augmentation for radiotherapy treatment applications. The primary dataset to be used(AD) has been taken from an anonymised sample of Alzheimer's disease patients who have undergone radiotherapy treatment. For the exploration of various GNN approaches, two different publicly available datasets (ADNI and CNI) will be used to explore the possible benefits of using a related dataset. In the following paragraphs, each dataset will be explored more thoroughly.

3.1.1 AD Dataset

This was the main dataset used for learning the baseline for experimentation and evaluation. Provided by researchers at Cambridge University Neuroscience Department, the data consisted of 202 2D DTI scans of the brain images with clear demarcation. It consists of 202 brain network datasets which include 101 Alzheimer's disease patients' brain data and 101 Control Normal brain data. All 202 patients and normal people had the scans taken while in the supine position. The produced images had consistent (x = 90) and (y = 90) dimension values. As for the physiology brain regions of the images, these are clearly shown on the images but not identified in the graph form. Usually, the brain DTI cannot be set as the input of the graph classification models. Then a classical data pre-processing method is required for transferring the brain DTI to brain atlas regions.

3.1.2 ADNI Dataset

The Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset is a publicity dataset that was implemented to improve clinical trials for prevention and treatment. This dataset was provided under the leadership of Dr. Micael W. Weiner, funded as a private-public partnership with 27 million contributed by 20 companies and two foundations through the Foundation for the National Institutes of Health and 40 million from the NIA. The initial five-year study (ADNI-1) was extended twice, which generated the ADNI-2 and ADNI-3 datasets. Thus, the ADNI dataset was given more expectations, such as detecting the earliest signs of AD, tracking and validating biomarkers for clinical AD trials, and universalizing all data for worldwide clinical trial designers and scientists.

Through these years, the research based on the ADNI dataset has achieved many significant results about how biomarkers become abnormal in the following order: β-amyloid, Tau, Glucose metabolism, structural MRI, and cognitive impairment. In our case, these discoveries combining deep learning algorithms have significantly increased the performance of the Alzheimer's disease classification task by over 90

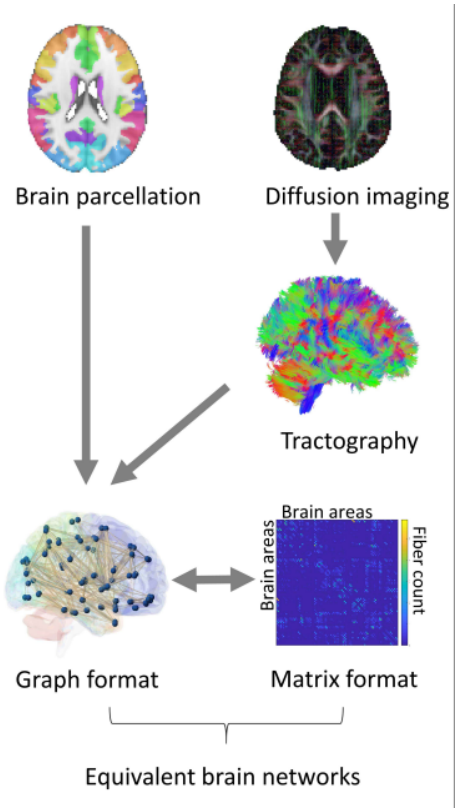

Figure 3.1: AD Dataset Example[LWC21]

In this project, we have prepared 218 brain DTI consisting of 109 Alzheimer's Disease and 109 Control Normal, which is also originally in the form of 2D scan images. The ADNI brain images also had consistent (x =90) and (y = 90) dimension values. The pre-processing method of ADNI is almost exactly the same as the pre-processing method of the AD dataset below.

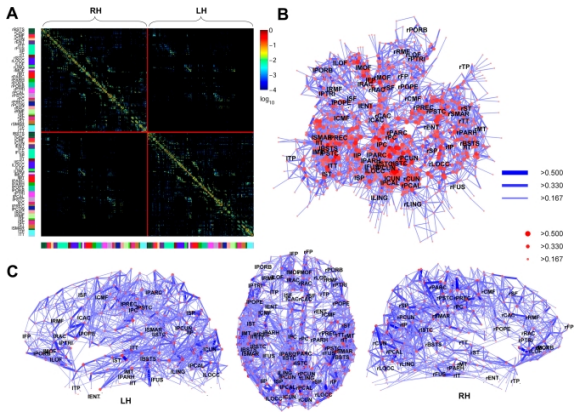

Figure 3.2: ADNI Dataset Example[HCG+08]

3.1.3 CNI Dataset

The CNI dataset or resting-state fMRI (rsfMRI) dataset is mainly proposed by the CNI Challenge 2019. The main idea of the challenge is trying to capture biologically relevant and generalisable information about the brain effectively. Before the challenge, most of the big open source datasets, such as the Human Connectome Project (HCP) and the Autism Brain Imaging Data Exchange (ABIDE), have just overfitted the brain data.

The CNI dataset consists of two categories, attention deficit hyperactivity disorder (ADHD) and neurotypical controls (NC). This challenge also requires a power graph classification model to predict subject diagnosis (ADHD vs. Neurotypical Control) based on the brain network data. The CNI dataset has 120 cases for each type. Each case have consistent (x=116) and (y=116) dimension values.

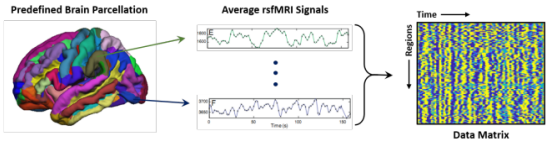

Figure 3.3: CNI Dataset Pipeline[SVR+21]

3.2 General Preprocessing

As mentioned above, the DTI brain images cannot be directly set as the input for neural graph networks. In this case, we have to apply a classical pre-processing method to transfer the DTI brain images to the brain connectivity matrix. The general connectivity matrix offers a compact description of the pairwise connectivity between all network nodes. To build a connectivity matrix, C, for a brain network comprising N nodes, we start by constructing a two-dimensional array called a square matrix, which comprises N rows and N columns. For a brain connectivity matrix, there are five pre-processing steps, as shown in 3.4.

1. Use the FSL (FMRIB software library) toolbox to pre-process the Brain diffusion MRI first.
2. Using the Diffusion Toolkit to capture the Tractography of the pre-processed Brain diffusion MRI.
3. Using the Automated Anatomical Labelling (AAL) atlas toolbox to divide the grey matter regions of Brain diffusion MRI into 90 brain regions.
4. Count the number of tracts between each pair of brain regions to produce the adjacency matrices. Connect those adjacency matrices to construct the structural brain networks.
5. Normalizing the track counts of the brain network between 0 and 1

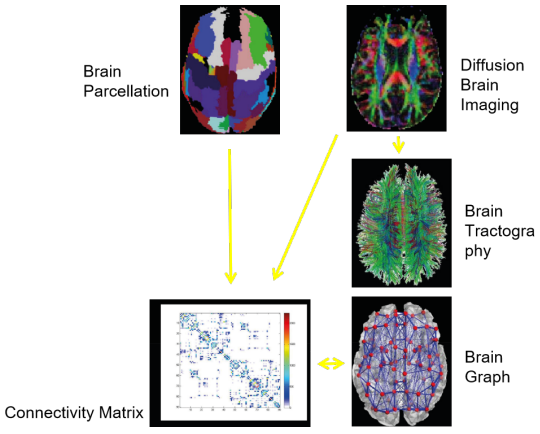

Figure 3.4: Brain Network Data Pre-processing Method

Chapter 4

Evaluation

In the following sections, various metrics will be explored that we have used in both the classification and augmentation tasks.

4.1 Classification Metrics

The classification metrics are standard for the evaluation of learning approaches. It is mainly composed of four elements, True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). For a binary classification task, True Positive (TP) indicates that the actual label and the predicted label are both positive and correct. False Positive (FP) indicates that the predicted label is positive, but the actual label is negative. True Negative (TN) indicates that both the actual and predicted labels are negative. False Negative (FN) indicates that the predicted label is negative, but the actual label is positive. These four elements are composed to form the classification task metrics.

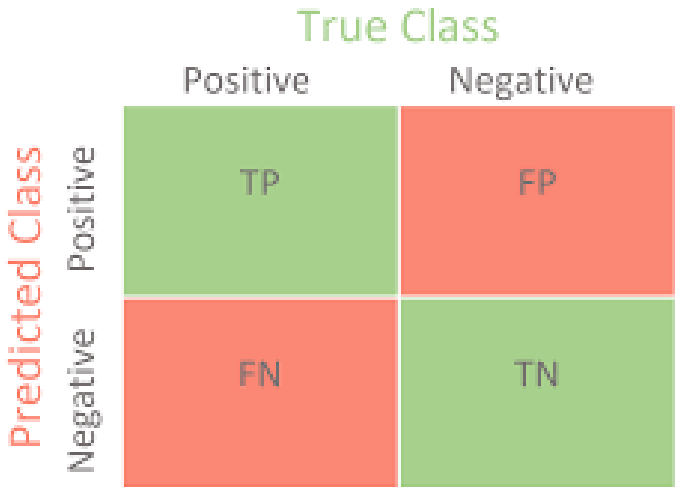

Figure 4.1: The Confusion Matrix

4.1.1 Precision

The Precision metrics indicate the possibilities of corrected labeled data when their predicted label is positive. In our case, it represents the possibility of corrected labeled data when the data is predicted to be Alzheimer's Disease. The range of the results is $[0, 1]$. The closer to 1, the better the results. The closer to 0, the worse the results.

In mathematically, Precision $= \frac{TP}{TP+FP}$.

4.1.2 Recall

The Recall metrics indicate the possibilities that data with positive labels are correctly predicted. In other words, the Recall metric is a measure of coverage. The measure has multiple positive examples and is divided into positive examples.

In mathematically, Recall = $\frac{\text{TP}}{\text{TP+FN}}$.

4.1.3 F-1 Score

The F-1 Score metrics indicate the average Precision and Recall. Sometimes, the precision and recall metrics appear contradictory. So, it is necessary to consider both of them comprehensively. The most common method is F-Measure (also known as F-Score).

The F-Measure is Precision and Recall weighted harmonic mean: $\text{F} = \frac{(\alpha^2+1)*\text{Precision}*\text{Recall}}{\alpha^2(\text{Recall}+\text{Precision})}$.

When $\alpha = 1$, it is known as the F-1 Score. $\text{F-1} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}*\text{Recall}}$

4.1.4 Accuracy

The accuracy rate is our most common evaluation metric, and it is easy to understand that it is the number of pairs of samples divided by the number of all samples. Generally speaking, the higher the accuracy rate, the better the classifier. Accuracy is indeed a good and intuitive evaluation indicator, but sometimes high accuracy does not represent an algorithm. For example, for predicting an earthquake on a specific day in a particular area, suppose we have a bunch of features as the attributes of earthquake classification, and there are only two categories: 0: no earthquake occurs, 1: earthquake occurs. A non-thinking classifier that divides the class into 0 for each test case, then it may achieve %99 accuracy, but when the real earthquake comes, the classifier is unaware; this classification brings A considerable loss. Why is a %99 accurate classifier not what we want? Because the data distribution here is not balanced, there is too little data in category 1, and a wholly misclassified category 1 can still achieve a high accuracy rate but ignore what we care about. Let us take another example. In the case of unbalanced positive and negative samples, the evaluation index of accuracy has a significant defect. For example, in Internet advertisements, the number of clicks is minimal, generally only a few thousandths. If accuracy is used, even if all the predictions are pessimistic (no clicks), accuracy will still have more than %99, which is meaningless. Therefore, it is far from scientific and comprehensive to evaluate an algorithm model solely by its accuracy.

In mathematically, Accuracy = $\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$

4.2 Data Augmentation Metrics

4.2.1 Kullback–Leibler divergence

In probability theory or information theory, KL divergence (Kullback–Leibler divergence), also known as relative entropy, is a way to describe the difference between two probability distributions, P and Q. It is asymmetric, which means D(P||Q) $\neq$ D(Q||P). In information theory, D(P||Q) represents the information loss when fitting the true distribution P with the probability distribution Q, where P represents the actual distribution and Q represents the fitted distribution of P. Some people call KL divergence KL distance, but in fact, KL divergence does not satisfy the concept of distance; it should be: 1) KL divergence is not symmetrical; 2) KL divergence does not satisfy the triangle inequality. For a discrete random variable or two probability distributions P and Q of a continuous random variable, the KL divergence is defined as follows, respectively.[Csi75]

For discrete random variable: $D(P||Q) = \sum_{i \in X} P(i) * [\log(\frac{P(i)}{Q(i)})]$.

For continuous random variable: $D(P||Q) = \int_x P(x) * [\log(\frac{P(i)}{Q(i)})]dx$

4.2.2 Maximum mean discrepancy

The Maximum mean discrepancy was first proposed for the two-sample test problem, which is used to judge whether the two distributions p and q are the same. Its basic assumption is: if for all functions f that takes the sample space generated by the distribution as input, if the means of the corresponding images on f of enough samples generated by the two distributions are equal, then

it can be considered that these two The distribution is the same distribution. It is now generally used to measure the similarity between two distributions.[GBR+12]

Specifically, the statistical test method based on MMD (maximize mean discrepancy) refers to the following method: based on samples from two distributions, by finding a continuous function f on the sample space, the function value of samples from different distributions on f is obtained. The mean discrepancy of the two distributions corresponding to f can be obtained by taking the difference between the two means. Find an f such that the mean discrepancy has the maximum value, and the MMD is obtained. Finally, MMD is taken as the test statistic to judge whether the two distributions are the same. If this value is small enough, the two distributions are considered to be the same. Otherwise, they are considered different. At the same time, this value is also used to judge the similarity between the two distributions. If F is used to denote a continuous function set in the sample space, then MMD can be expressed by the following formula:

$\mathrm{MMD}[\mathcal{F}, p, q] = \sup(E_{x\sim p}[f(x)] - E_{y\sim q}[f(y)])$

Chapter 5

Methodologies

In this chapter, we have explored and explained some of the main areas of research in the domain of Alzheimer's Disease classification field. Before the Graph Neural Network was proposed, most of the classification tasks were accomplished by the end-to-end convolution neural network. In our case, the BrainNet CNN is the most related work that applies a specific convolution layer to improve the performance of the Alzheimer's Disease classification task. However, CNN can often be an unhittable approach due to the characterisation of the Brain Network data. Therefore, we present the various graph neural network models evaluated in this research, hypothesize the outcomes we expect to see as well as summarise the results of our evaluation which follows in more detail in the next chapter.

We begin this chapter by first exploring the baseline CNN approaches chosen as benchmark comparisons for different datasets (5.1). As mentioned before, we believe GNNs are more suitable for handling the brain connectivity matrix data. Then we present the various GNN methods with different node features (5.2). Following this exploration, we investigate spatiality as a technique to use the topological feature and GNN to verify its potential (5.3).

5.1 Convolution Neural Network

As one of the most widely applied models in the deep learning field, a convolutional neural network (CNN) is a stacking of artificial neural networks (ANN). It was implemented for several tasks, such as image recognition, natural language processing, and disease classification, in our case.[VNL+20]

Strictly, CNN is regularized version of multilayer perceptrons (MLP), which can also be interpreted as fully connected networks. It states that each neuron in one layer of CNN is connected to all neurons in the next layer. Although this classical architecture effectively reduced the occupied memory by sharing the weights, it made them prone to overfitting data. Thus, some overfitting prevention methods, such as penalizing parameters, cross-validation method, and trimming connectivity, are necessary for the training process of CNNs. [GBC16]

In detail, there are three critical operations of the convolutional neural network, one is the local receptive field, the other is weight sharing, and the third is the pooling layer, which effectively reduces the number of parameters of the network and alleviates the overfitting problem of the model. The convolutional layer and pool sampling layer of the hidden layer is the core modules to realize the feature extraction function of the convolutional neural network. The network model uses the gradient descent method to minimize the loss function to reversely adjust the weight parameters in the network layer by layer and improves the accuracy of the network through frequent iterative training. The lower hidden layer of the convolutional neural network is composed of alternate convolutional layers and max pooling sampling layers. The upper layer is the fully connected layer corresponding to the hidden layer and logistic regression classifier of the traditional multilayer perceptron. And the input of the first fully connected layer is the feature image obtained by feature extraction by the convolutional layer and the subsampling layer. The last output layer is a classifier using logistic regression, Softmax regression, or even a support vector machine to classify the input image.[AW18]

Usually, the convolutional neural network structure includes a convolutional layer, downsampling layer, and full connection layer. Each layer has multiple feature maps, each feature map extracts a feature of the input through a convolution filter, and each feature map has multiple neu-

rons. After the input image statistics and filters are convolved, the local feature is extracted. Once the local feature is extracted, its positional relationship with other features is also determined. The input of each neuron is the same as that of the previous layer. The local receptive fields are connected, and each feature extraction layer is followed by a computing layer for local averaging and secondary extraction, also called the feature mapping layer. Each computing layer of the network consists of multiple feature mapping planes. The weights of the neurons are equal. Then, the mapping from the input layer to the hidden layer is called a feature map. The feature extraction layer is obtained through the convolution layer, and the feature mapping layer is obtained after pooling.

5.1.1 BrainNet CNN

BrainNet CNN is the most classical convolutional neural network for handling the brain network dataset. It is also set as the primary discriminator and classifier in the BrainNet GAN. [LWC21]. In this case, we treated the BrainNet CNN as the CNN benchmark for the Alzheimer's Disease classification task. BrainNet CNN consists of novel edge-to-edge, edge-to-node, and node-to-graph convolutional filters that effectively exploit the topological locality of structural brain networks compared to the traditional image-based CNNs.

An edge-to-edge (E2E) layer is similar to a standard convolutional layer on grid-like data in a CNN because it filters the data locally. In mesh-like data, filters can be defined according to spatial locality, while E2E filters can be defined according to the topological locality, combining the weights of edges that share nodes.

Mathematically, let $G^{l,m} = (A^{l,m}; \omega)$ represent the m-th feature map of a weighted brain network at the l-th layer of the BrainNet CNN. Then the edge-to-edge (E2E) layer can be interpreted as follows:

$A_{i,j}^{l+1,n} = \sum_{m=1}^{M^l} \sum_{k=1}^{|\omega|} r_k^{l,m,n} A_{i,k}^{l,m} + c_k^{l,m,n} A_{k,j}^{l,m}$, where $A^{l,m} \in R^{|\omega|*|\omega|}$ is the adjacency matrix of brain network data, ω represents the set of brain nodes, and $[c^{l,m,n}, r^{l,m,n}] = w^{l,m,n} \in R^{2|\omega|}$. The $w^{l,m,n}$ is the trainable weight.

As for the edge-to-node (E2N) filter, it also takes an adjacency matrix, $A^{l,m}$ from each feature map as input and outputs a vector of size $|\omega|$. Then the E2N layer can be represented as follows: $a_i^{l+1,n} = \sum_{m=1}^{M^l} \sum_{k=1}^{|\omega|} r_k^{l,m,n} A_{i,k}^{l,m} + c_k^{l,m,n} A_{k,i}^{l,m}$, where $[c^{l,m,n}, r^{l,m,n}] = w^{l,m,n} \in R^{2|\omega|}$. The $w^{l,m,n}$ is the trainable weight.

Finally, similar to the E2N layer, the node-to-graph (N2G) layer reduces the dimensionality of the input, in this case by weighted combinations of nodes to output a single scalar,

$a_i^{l+1,n} = \sum_{m=1}^{M^l} \sum_{i=1}^{|\omega|} w_i^{l,m,n} a_i^{l,m}$. [KBM+16]

5.2 Graph Neural Network and Node features

Unsimilar to the learning process of CNNs, the performance of GNNs still highly depends on the hand-designed node features of the input graphs. The handed-designed node features can effectively affect the performance of the GNN since node features are the most essential element in characterising the structure and position of a node in the network. Moreover, most GNNs take the node feature as the essential factor in understanding a whole graph. In our case, the choice of hand-designed features can effectively affect the final performance of the Alzheimer's Disease classification task. Thus, it is undoubted that we have to try many different hand-designed node features as the input of GNNs to improve the accuracy of the classification task.

5.2.1 Isomorphic Graph

An isomorphic graph means only one node type and one relationship type in the graph. The isomorphic graph is the simplest case of the actual graph data, such as the World Wide Web, composed of hyperlink relationships, and the information of this kind of graph data is all contained in the adjacency matrix.[GJ79]

In our case, the brain connectivity matrix is naturally an isomorphic graph since the different atlas brain regions cannot be weighted. Thus, we treat all the node features of the brain connectivity matrix as 1.

5.2.2 Node Degree

The node degree feature represents the number of edges connected to the node. For any adjacency matrix A, the node degree of a node indexed by i in a graph is $k_i = \sum_j a_{ij}$, where the sum is over all nodes in the network. [WZY17] However, the node degree counts the neighboring nodes without capturing their importance.

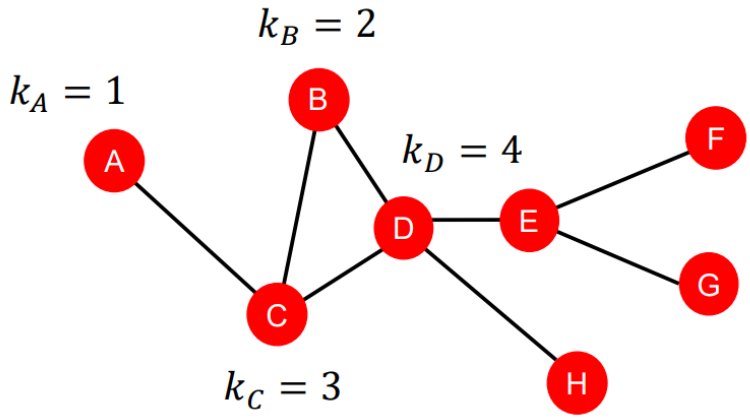

Figure 5.1: Node Degree

5.2.3 Node Strength

Node strength considers both the importance of nodes and their neighboring edges. For any weighted graphs, edges can be utilized to capture variations in the connectivity strength between pairs of nodes. In our case, the connectivity weights are measured as the number of reconstructed streamlines linking pairs of the diffusion MRI brain regions.[FZB16]

Mathematically, the node strength can be represented as follows:

$s_i = \sum_{j \neq i} w_{ij}$ where s_i denotes the node strength and w_{ij} represents the weight of edges that connect node i and node j. [BBV07]

5.2.4 Eigenvector Centrality

The eigenvector centrality measures the importance of a node in a graph. In detail, a node v is important if it is surrounded by important neighboring nodes $u \in N(v)$. Then we can model the eigenvector centrality of node v as the sum of the centrality of neighboring nodes, $c_v = \frac{1}{\lambda} \sum_{u \in N(v)} c_u$, where λ is the normalisation constant. The function can be rewritten as follows: $\lambda c = Ac$ where A is the adjacency matrix, c is the centrality vector, and λ is the eigenvalue. So, the centrality c is, in fact, the eigenvector of A. Thus, relative scores are assigned to all nodes of a graph based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. A high eigenvector score means that a node is connected to many nodes that have high scores. [ZMJ14]

5.2.5 Betweeness Centrality

The betweenness centrality is a measure of centrality in a graph based on the shortest paths. For every pair of nodes in a connected graph, there must exist more than one shortest connection between them. In this way, either the number of edges that the path passes through or the sum of the edge weights is minimized. Thus, the betweenness centrality for each node is the number of shortest paths that are connected with them. It is further stated that a node is crucial if it lies on many shortest paths between other nodes.[Fre77]

Mathematically, the betweenness centrality can be represented as follows:

$c_v = \sum_{i \neq v \neq j} \frac{\text{number of shortest paths between i and j that contain v}}{\text{number of shortest paths between i and j}}$

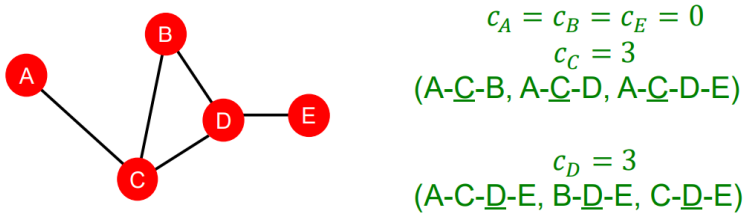

Figure 5.2: Betweenness Centrality

5.2.6 Clustering Coefficient

The clustering coefficient measures the degree to which nodes in a graph tend to be in the same group. It is mainly used for real-world networks such as social networks because the nodes in the social networks are more likely to create tightly knit groups characterised by a relatively high density of ties. [DGK05]

In particular, the clustering coefficient measures how connected $v's$ neighboring nodes are: $e_v = \frac{\text{number of edges among neighboring nodes}}{k_v} \in [0, 1]$

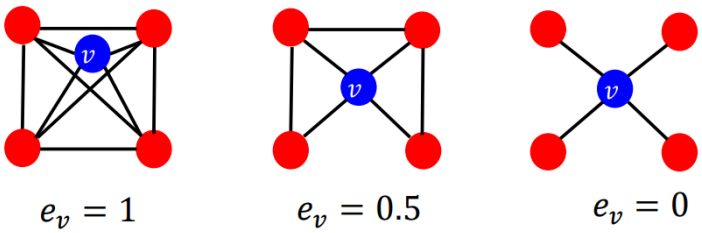

Figure 5.3: Clustering Coefficient

5.3 Spatiality Computation

The graph neural network can be divided into spectral graph neural networks and spatial graph neural networks. The spectral graph neural network such as GCN and GIN is capable of most of the graph classification tasks because most graph does not have stable node positions. However, it is necessary to consider the spatial information of the brain connectivity matrix since the coordinates of nodes (brain atlas regions) are always stable. Moreover, considering the spatial information can also effectively capture the topological and structural features of the brain connectivity matrix, which improves the Alzheimer's Disease classification task.

5.3.1 Spatial Graph Convolution Network

For any data structure, the data can be expressed as a graph, $G = (V, E, A)$ which contains a set of nodes $v1, v2, ..., v_n \in V$, a set of edges $e_{v_i,v_j} \in E$ and adjacency matrix $A = [a_{ij}]_{i,j=1}^n$. Nodes V represent the graph's vertices and edges E represent the connections between the vertices. Usually, if there is a connected edge from v_i to v_j, then $a_{ij} = 1$. If $a_{ij} = 0$, it means there is no connection between nodes v_i and v_j. This connection information can be summarized as the corresponding adjacency matrix, A. [SGT+09] In our case, the nodes v_i and v_j in the brain network can represent as brain atlas region. The edge e_{v_i,v_j} indicates the connections between different brain atlas regions.

Unlike the traditional graph convolution network, the spatial convolution network usually assumes that each node v_i is characterised by its coordinates $p_i \in R^t$. Then for any images, p_i can be interpreted as a vector of 2-D coordinates. In our case, p_i demonstrates the location of brain atlas regions in 3-D space. Thus, p_i is invariant across layers since the location of atlas regions cannot be simply changed.

Mathematically, the spatial convolution operation can be interpreted as follows: $h_i(U, b) = \sum_{j \in N_i} \text{ReLU}(U^T(p_j - p_i) + b) \odot h_j$, where $U \in R^{t \times d}$, $b \in R^d$ are the trainable parameters, d

denotes the dimension of h_j and $\odot$ is the element-wise multiplication. These trainable parameters U, b also operate on the neighborhood of a node v_i in the spatial convolutional filter. [DST+20]

5.3.2 Spatial Node Feature

To improve the performance of the classification task, we proposed a novel spatial node feature of Brain Connectivity Matrix, which combines both superiorities of spatial convolution and graph neural network.

For any data structure, the data can be expressed as a graph, $G = (V, E, A)$ which contains a set of nodes $v1, v2, ..., v_n \in V$, a set of edges $e_{v_i,v_j} \in E$ and adjacency matrix $A = [a_{ij}]_{i,j=1}^{n}$. Nodes V represent the graph's vertices and edges E represent the connections between the vertices. Usually, if there is a connected edge from v_i to v_j, then $a_{ij} = 1$. If $a_{ij} = 0$, it means there is no connection between nodes v_i and v_j. This connection information can be summarized as the corresponding adjacency matrix, A. [SGT+09] In our case, the nodes v_i and v_j in the brain network can represent as brain atlas region. The edge e_{v_i,v_j} indicates the connections between different brain atlas regions.

As mentioned above, a typical graph convolution is defined by combing two operations. In mathematically, a graph convolution for each node v_i and its neighboring feature vectors $N_i = j : a_{ij} = 1$ can be interpreted as follows:

$\overline{h}_i = \sum_{j \in N_i} u_{ij} h_j$, where $h_i \in R^{d_{in}}$ are the node features.

In matrix form, it can be interpreted as follows: $\widetilde{H} = UH^T$, where $H = [h_1, .., h_n]$ denotes the matrix of node features and $U \in R^{n \times n}$ are the trainable weight.

Usually, the output of a given graph convolution layer is given by a standard multiple layer perceptron (MLP):

$\mathrm{MLP}(\overline{H}; W) = \mathrm{ReLU}(W^T \overline{H} + b)$, where $W \in R^{d_{in} \times d_{out}}$ is a trainable weight matrix and $b \in R^{d_{out}}$.

Then we assume that each node v_i is characterised by its coordinates $p_i \in R^3$ for any brain connectivity matrix. We are taking the location of any nodes (brain atlas regions) as the node feature that participated in the graph convolution computation process. Mathematically, let $H = [h_x, h_y, h_z]$ denote the node features of a graph, where h_x, h_y, h_z represents the coordinates of a node.

However, taking the spatial coordinates as the only node feature ignores the importance of edge weight and nodes' neighboring information, which may even decrease the classification task performance. In this case, it is necessary to add more hand-designed node features, such as node strength, to consider more information on a graph. Thus, we have utilized a novel spatial node feature $H = [h_x, h_y, h_z, h_{strength}]$ as the input of graph neural networks, where $H \in R^4$ and $h_{strength}$ denotes the node strength of a graph.

Since the node strength can be interpreted as follows: $s_i = \sum_{j \neq i} w_{ij}$ where s_i denotes the node strength and w_{ij} represents the weight of edges that connect node i and node j. [BBV07]. Then it is a necessary supplementary node feature for our implementation.

Chapter 6

Results

In this chapter, we will report our detailed results for the performance of the several proposed methods discussed in the previous chapter. The results for each dataset are discussed separately. The structure of analysis for each dataset is consistent. Firstly, we present all the experimental set-up in the training process of different datasets. Then we analyze the results by presenting the quantitative comparison images between various proposed methods. Finally, we have provided conclusions that follow the analysis.

6.1 Preliminaries

In this work, we have utilized three different Brain MRI datasets, ADNI, CNI, and AD. We wish to utilize these three different Brain MRI datasets to verify the robustness and liability of the GNN and the novel spatial node feature. Additionally, each method is executed several times in separate training contexts. In this way, we have avoided the randomness of the results. Moreover, we are also tuning the hyperparameters to get better results during the training period.

6.2 ADNI

The Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset was effectively processed for analyzing the pathological regions of the braBrainI. In this case, GNNs and CNN perform great on the ADNI dataset, as shown in the table 6.2. The red number on the table represents the highest accuracy in a row. It clearly states that the best performance of the classification task happens when the node feature is the node degree. Moreover, the traditional CNN approach and spatial GCN also perform well on the ADNI dataset.

6.2.1 CNN vs. GNN

As mentioned above, it is necessary to utilize the GNNs to capture the topological and structural features of the diffusion MRI. Before GNN was presented, most of the research treated the diffusion Brain MRI as a matrix representation that ignores the topological feature. In this case, transforming the brain MRI data into graph form is necessary to compute its topological and structural features.

Alzheimer's Disease Neuroimaging Initiative (ADNI)																		
	1		Degree		Strength		4D Strength		Eigenvector		Betweenness		Clustering		Matrix		Location	
	Train	*Test*	*Train*	*Test*	*Train*	*Test*	*Train*	*Test*	*Train*	*Test*	*Train*	*Test*	*Train*	*Test*	*Train*	*Test*	*Train*	*Test*
GNN	0.5333	0.4265	0.8533	0.8676	0.8800	0.8382	0.9200	0.8676	0.4733	0.5588	0.4733	0.5588	0.9600	0.9412				
GCN	0.5333	0.4265	0.9133	0.9559	0.9467	0.9265	0.9800	0.9118	0.5267	0.4412	0.5267	0.4412	0.9200	0.8971				
GIN	0.9000	0.9559	0.9133	0.9559	0.9067	0.9118	0.9933	0.9412	0.9933	0.9412	0.9800	0.9265	0.9533	0.9265				
Graclus	0.8400	0.8824	0.9600	0.9853	0.9600	0.9412	0.9933	0.9412	0.8533	0.8529	0.8600	0.8382	0.9467	0.9853				
TopK	0.8400	0.8824	0.9200	0.9853	0.9600	0.9265	0.9933	0.9706	0.8800	0.9265	0.8600	0.8382	0.9467	0.9412				
SAG Pool	0.9333	0.9853	0.9333	0.9853	0.9933	0.9265	0.9733	0.9706	0.9067	0.8971	0.9000	0.8529	0.9533	0.9706				
Edge Pool	0.9400	0.9706	0.9600	0.9853	0.9733	0.9559	0.9867	0.9706	0.9600	0.9265	0.9667	0.9412	0.9333	0.9853				
Graph Attention Network	0.8400	0.8824	0.8600	0.9168	0.9800	0.9265	0.9467	0.9118	0.8533	0.8529	0.8600	0.8382	0.9467	0.9853				
Set2Set	0.8400	0.8824	0.9067	0.9412	0.9800	0.9412	0.9933	0.9265	0.8533	0.8529	0.5267	0.4412	0.8267	0.8235				
Sort Pool	0.8800	0.9412	0.8667	0.8971	0.9133	0.8529	0.9867	0.8971	0.9067	0.8824	0.9133	0.8529	0.9267	0.9265				
ASAP	0.5333	0.4265	0.9267	0.9853	0.9600	0.9265	0.9200	0.9706	0.5267	0.4412	0.9733	0.9412	0.9533	0.9265				
BrainNet CNN															0.9933	0.9853		
Spatial GCN																	0.9733	0.9412

Table 6.1: ADNI Results

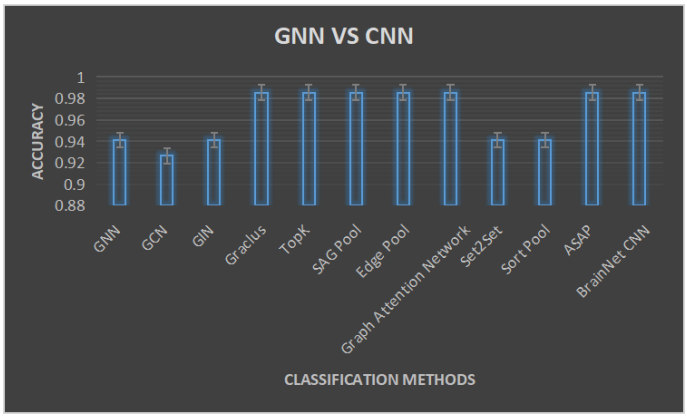

Figure 6.1: Comparison between CNN and GNN

Alzheimer's Disease Neuroimaging Initiative														
	1		Degree		Strength		4D Strength		Eigenvector		Betweenness		Clustering	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
GNN	0.5333	0.4265	0.8533	0.8676	0.8800	0.8382	0.9200	0.8676	0.4733	0.5588	0.4733	0.5588	0.9600	0.9412
GCN	0.5333	0.4265	0.9133	0.9559	0.9467	0.9265	0.9800	0.9118	0.5267	0.4412	0.5267	0.4412	0.9200	0.8971
GIN	0.9000	0.9559	0.9133	0.9559	0.9067	0.9118	0.9933	0.9412	0.9933	0.9412	0.9800	0.9265	0.9533	0.9265
Graclus	0.8400	0.8824	0.9600	0.9853	0.9600	0.9412	0.9933	0.9412	0.8533	0.8529	0.8600	0.8382	0.9467	0.9853
TopK	0.8400	0.8824	0.9200	0.9853	0.9600	0.9265	0.9933	0.9706	0.8800	0.9265	0.8600	0.8382	0.9467	0.9412
SAG Pool	0.9333	0.9853	0.9333	0.9853	0.9933	0.9265	0.9733	0.9706	0.9067	0.8971	0.9000	0.8529	0.9533	0.9706
Edge Pool	0.9400	0.9706	0.9600	0.9853	0.9733	0.9559	0.9867	0.9706	0.9600	0.9265	0.9667	0.9412	0.9333	0.9853
Graph Attention Network	0.8400	0.8824	0.8600	0.9168	0.9800	0.9265	0.9467	0.9118	0.8533	0.8529	0.8600	0.8382	0.9467	0.9853
Set2Set	0.8400	0.8824	0.9067	0.9412	0.9800	0.9412	0.9933	0.9265	0.8533	0.8529	0.5267	0.4412	0.8267	0.8235
Sort Pool	0.8800	0.9412	0.8667	0.8971	0.9133	0.8529	0.9867	0.8971	0.9067	0.8824	0.9133	0.8529	0.9267	0.9265
ASAP	0.5333	0.4265	0.9267	0.9853	0.9600	0.9265	0.9200	0.9706	0.5267	0.4412	0.9733	0.9412	0.9533	0.9265

Table 6.2: ADNI Node Feature Comparison Results

As shown in the figure 6.1, BrainNet CNN performs well on the ADNI dataset. The GNN also states its availability to handle the Brain MRI dataset. For example, the Graclus, TopK pooling, SAG pooling, Edge pooling, and graph attention network have shown superiority in accomplishing the Alzheimer's Disease classification task.

6.2.2 GNN Node Feature Comparison

After we have shown the availability of utilizing GNN for accomplishing the Alzheimer's Disease classification, it is essential to test different node features to improve the performance as shown in the table 6.2, taking the node degree as the node feature reaches the highest performance. Otherwise, our novel spatial node feature also performs well on the ADNI dataset.

As shown in figure 6.2 and figure 6.3, it demonstrates that taking 1 as the node feature performed the lowest accuracy since the homogeneous graph does not demonstrate any vital node and edge information. Most of the centrality node features, such as the Betweenness centrality and clustering coefficient, have shown insufficient robustness because they performed well on some classification models but performed poorly on others. For our novel spatial node feature, the nearly horizontal line chart clearly demonstrates its robustness. Although it may not perform best all the time, it also reaches top performance compared to other node features.

6.2.3 Spectral GNN vs. Spatial GCN

As mentioned before, graph neural networks are divided into spectral GNN and spatial GCN. In this case, it is necessary to compare them on the Alzheimer's Disease classification task. It took the spatial node feature as the input for all spectral neural networks and compared the results between spectral GNN and spatial GCN, as shown in figure 6.4. It shows that our work performs better on many classification methods since it combines both topological features and spatial features.

6.2.4 Summary

In summary, our novel spatial node feature performs excellently on the ADNI datasets. Although it may not reach the state-of-the-art, it also reaches over 90% accuracy. Moreover, it also clearly

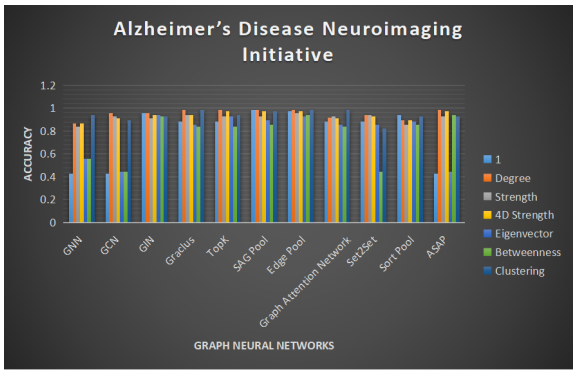

Figure 6.2: Histogram of ADNI Node Feature Comparison Results

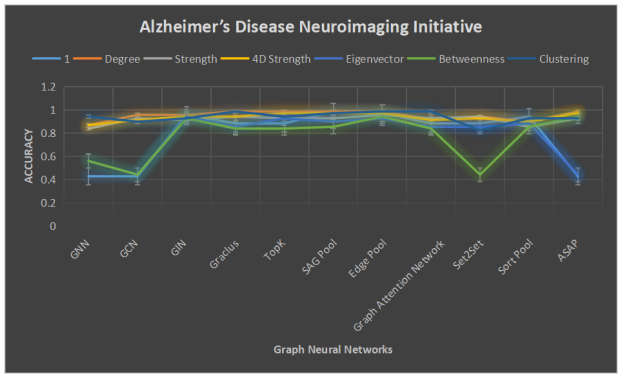

Figure 6.3: Line Chart of ADNI Node Feature Comparison Results

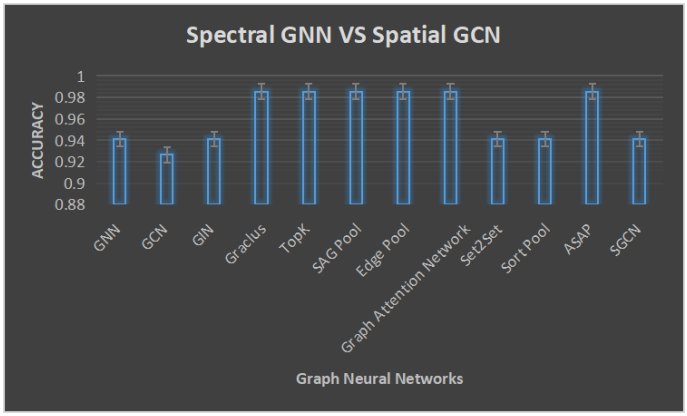

Figure 6.4: Comparison between Spectral GNN and Spatial GCN

demonstrates its robustness compared to other node centrality features such as Betweenness centrality and clustering coefficient. Thus, these results prove that our expectations, which aggregate both structural and spatial information, have been achieved.

6.3 CNI

For the CNI or rsfMRI dataset, it is mainly implemented for functional MRI. We wish to test the generalisability of the GNNs, to see the performance of the functional MRI classification task. As shown in the table 6.3, both GNNs and CNN do not perform well on the CNI dataset. The red number on the table represents the highest accuracy in a row. It clearly states that the best performance of the classification task happens when the node feature is the node strength. Moreover, the traditional CNN approach performs best on the CNI dataset. However, our novel spatial node feature was not utilized on the CNI dataset since the coordinates of the brain nodes were not provided.

6.3.1 CNN vs. GNN

As mentioned above, it is necessary to utilize the GNNs to capture the topological and structural features of the functional MRI. Before GNN was presented, most of the research treated the functional BraBrainI as a matrix representation that ignores the topological feature. In this case, transforming the brain MRI data into graph form is necessary to compute its topological and structural features.

c

As shown in the figure **??**, BrainNet CNN performs well on the CNI dataset. The GNN also states its availability in handling the functional Brain MRI dataset, and GCN even reaches the

CNI														
	1		Degree		Strength		Eigenvector		Betweenness		Clustering		Matrix	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
GNN	0.5133	0.4600	0.5067	0.4833	0.5933	0.5800	0.5000	0.5000	0.5067	0.4800	0.5000	0.5000		
GCN	0.7467	0.6400	0.7733	0.5600	0.6467	0.5000	0.7600	0.5400	0.6067	0.7200	0.7200	0.5400		
GIN	0.4867	0.5400	0.5067	0.4800	0.6067	0.5600	0.5000	0.5000	0.5067	0.5200	0.5000	0.5000		
Graclus	0.5133	0.4600	0.5067	0.4800	0.6800	0.5200	0.5000	0.5000	0.5067	0.4800	0.5000	0.5000		
TopK	0.6000	0.5600	0.5867	0.5600	0.7133	0.5800	0.5000	0.5000	0.5067	0.4800	0.6067	0.5400		
SAG Pool	0.5933	0.5800	0.6133	0.5800	0.7600	0.6200	0.5000	0.5000	0.5067	0.4800	0.6267	0.5400		
Edge Pool	0.6267	0.5600	0.5600	0.6200	0.6000	0.6200	0.5000	0.5000	0.5067	0.4800	0.6000	0.5400		
Graph Attention Network	0.5133	0.4600	0.5067	0.4800	0.5933	0.5600	0.5000	0.5000	0.5067	0.4800	0.5000	0.5000		
Set2Set	0.5133	0.4600	0.5067	0.4800	0.5933	0.5400	0.5000	0.5000	0.5067	0.4800	0.5000	0.5000		
Sort Pool	0.5133	0.4600	0.5067	0.4800	0.9133	0.5600	0.5000	0.5000	0.5067	0.4800	0.5000	0.5000		
ASAP	0.6067	0.5800	0.6000	0.5600	0.7000	0.5800	0.6067	0.5800	0.5400	0.5400	0.6067	0.5200		
BrainNet CNN													0.9933	0.6800

Table 6.3: CNI Results

CNI												
	1		Degree		Strength		Eigenvector		Betweenness		Clustering	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
GNN	0.5133	0.4600	0.5067	0.4833	0.5933	0.5800	0.5000	0.5000	0.5067	0.4800	0.5000	0.5000
GCN	0.7467	0.6400	0.7733	0.5600	0.6467	0.5000	0.7600	0.5400	0.6067	0.7200	0.7200	0.5400
GIN	0.4867	0.5400	0.5067	0.4800	0.6067	0.5600	0.5000	0.5000	0.5067	0.5200	0.5000	0.5000
Graclus	0.5133	0.4600	0.5067	0.4800	0.6800	0.5200	0.5000	0.5000	0.5067	0.4800	0.5000	0.5000
TopK	0.6000	0.5600	0.5867	0.5600	0.7133	0.5800	0.5000	0.5000	0.5067	0.4800	0.6067	0.5400
SAG Pool	0.5933	0.5800	0.6133	0.5800	0.7600	0.6200	0.5000	0.5000	0.5067	0.4800	0.6267	0.5400
Edge Pool	0.6267	0.5600	0.5600	0.6200	0.6000	0.6200	0.5000	0.5000	0.5067	0.4800	0.6000	0.5400
Graph Attention Network	0.5133	0.4600	0.5067	0.4800	0.5933	0.5600	0.5000	0.5000	0.5067	0.4800	0.5000	0.5000
Set2Set	0.5133	0.4600	0.5067	0.4800	0.5933	0.5400	0.5000	0.5000	0.5067	0.4800	0.5000	0.5000
Sort Pool	0.5133	0.4600	0.5067	0.4800	0.9133	0.5600	0.5000	0.5000	0.5067	0.4800	0.5000	0.5000
ASAP	0.6067	0.5800	0.6000	0.5600	0.7000	0.5800	0.6067	0.5800	0.5400	0.5400	0.6067	0.5200

Table 6.4: CNI Node Features Comparison Results

highest performance. All these have proved the superiority of GNNs on brain disease classification tasks.

6.3.2 GNN Node Feature Comparison

After we have shown the availability of utilizing GNN for accomplishing the brain disease classification, it is essential to test different node features to improve the performance. As shown in the table 6.4, taking the node strength as the node feature reaches the highest performance. Otherwise, the Betweenness centrality node feature also reached the highest accuracy with the GCN model.

As shown in figure 6.5 and figure 6.6, it demonstrates that taking 1 as the node feature performed the lowest accuracy since the homogeneous graph does not demonstrate any vital node and edge information. For most of the node features, they performed stable on different classification models. However, none of their accuracies reached over 90%.

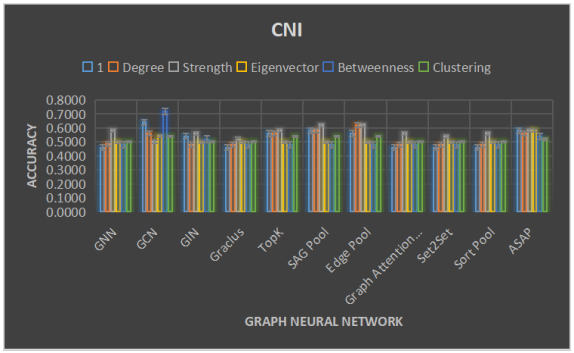

Figure 6.5: Histogram of CNI Node Feature Comparison Results

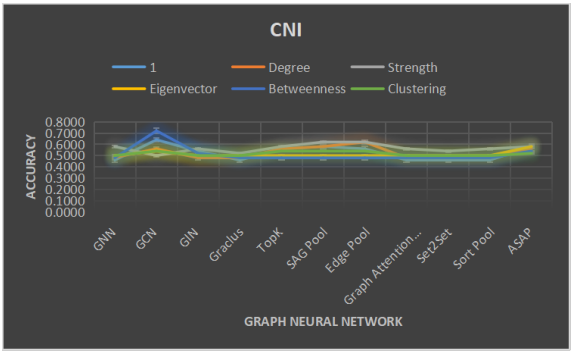

Figure 6.6: Line Chart of CNI Node Feature Comparison Results

6.3.3 Summary

In summary, GNNs are not only able to handle the diffusion MRI datasets but also able to handle the functional MRI datasets. However, none of these methods have reached over 90%, which means

Alzheimer's Disease														
	1		Degree		Strength		4D Degree		4D Strength		BrainNet CNN		SGCN	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
GNN	0.5067	0.5385	0.6400	0.6346	0.4867	0.5962	0.7600	0.6731	0.6333	0.5769				
GCN	0.5067	0.5385	0.6000	0.6154	0.5333	0.4231	0.9667	0.6154	0.9467	0.7115				
GIN	0.7133	0.5962	0.6067	0.6154	0.7267	0.6538	0.6467	0.6346	0.9800	0.5769				
Graclus	0.5067	0.5385	0.7200	0.6931	0.5200	0.4038	0.8067	0.6346	0.8200	0.6731				
TopK	0.5067	0.5385	0.5267	0.4808	0.5733	0.5000	0.8600	0.7115	0.8400	0.5962				
SAG Pool	0.5067	0.5385	0.5267	0.4808	0.4867	0.5962	0.9533	0.7308	0.8800	0.6154				
Edge Pool	0.5067	0.5385	0.5867	0.5385	0.5133	0.4038	0.7800	0.7115	0.5200	0.5000				
Graph Attention Network	0.5067	0.5385	0.5400	0.6346	0.5133	0.4038	0.6467	0.4615	0.6400	0.6923				
Set2Set	0.5067	0.5385	0.5267	0.4808	0.5067	0.4423	0.9600	0.6731	0.9933	0.7115				
Sort Pool	0.5067	0.5385	0.8267	0.6538	0.7067	0.6923	0.8533	0.6731	0.9000	0.6731				
ASAP	0.5067	0.5385	0.5867	0.5577	0.5200	0.4038	0.5267	0.6154	0.5200	0.5000				
BrainNet CNN											0.9067	0.6923		
SGCN													0.5533	0.4038

Table 6.5: AD Results

GNNs may not be reasonably sufficient in handling the function MRI datasets. Besides that, we cannot test the performance of spatial node features since the coordinates of the brain nodes are not given.

6.4 AD

The Alzheimer's Disease (AD) dataset was collected by Cambridge University, Neuroscience department. Both GNNs and CNN perform fine on the AD dataset, as shown in the table 6.5. The red number on the table represents the highest accuracy in a row. Before the guns were proposed, the state-of-the-art performance was acchieved by the CNN approach, BrainNet CNN. The table clearly shows that the BrainNet CNN performs best compared to other GNNs architectures. However, our novel spatial node feature (4D degree and 4D strength) clearly states that they outperformed both traditional CNN and GNN approaches.

6.4.1 CNN vs. GNN

As mentioned above, it is necessary to utilize the GNNs to capture the topological and structural features of the diffusion MRI. Before GNN was presented, most of the research treated the diffusion Brain MRI as a matrix representation that ignores the topological feature. In this case, transforming the brain MRI data into graph form is necessary to compute its topological and structural features.

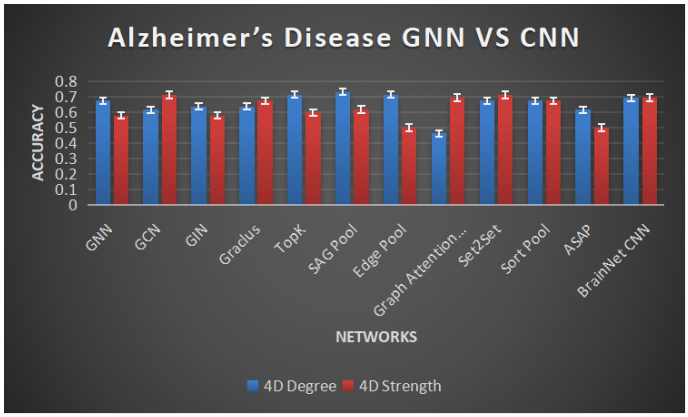

Figure 6.7: Comparison between CNN and GNN

As shown in the figure 6.10, BrainNet CNN performs well on the ADNI dataset. The GNN also states its availability to handle the Brain MRI dataset. For example, the Graclus, TopK pooling, SAG pooling, Edge pooling, and graph attention network have shown superiority in accomplishing the Alzheimer's Disease classification task. Moreover, the SAG pooling method with our novel spatial node feature has reached the state-of-the-art 73.08% accuracy.

Alzheimer's Disease										
	1		Degree		Strength		4D Degree		4D Strength	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
GNN	0.5067	0.5385	0.6400	0.6346	0.4867	0.5962	0.7600	0.6731	0.6333	0.5769
GCN	0.5067	0.5385	0.6000	0.6154	0.5333	0.4231	0.9667	0.6154	0.9467	0.7115
GIN	0.7133	0.5962	0.6067	0.6154	0.7267	0.6538	0.6467	0.6346	0.9800	0.5769
Graclus	0.5067	0.5385	0.7200	0.6931	0.5200	0.4038	0.8067	0.6346	0.8200	0.6731
TopK	0.5067	0.5385	0.5267	0.4808	0.5733	0.5000	0.8600	0.7115	0.8400	0.5962
SAG Pool	0.5067	0.5385	0.5267	0.4808	0.4867	0.5962	0.9533	0.7308	0.8800	0.6154
Edge Pool	0.5067	0.5385	0.5867	0.5385	0.5133	0.4038	0.7800	0.7115	0.5200	0.5000
Graph Attention Network	0.5067	0.5385	0.5400	0.6346	0.5133	0.4038	0.6467	0.4615	0.6400	0.6923
Set2Set	0.5067	0.5385	0.5267	0.4808	0.5067	0.4423	0.9600	0.6731	0.9933	0.7115
Sort Pool	0.5067	0.5385	0.8267	0.6538	0.7067	0.6923	0.8533	0.6731	0.9000	0.6731
ASAP	0.5067	0.5385	0.5867	0.5577	0.5200	0.4038	0.5267	0.6154	0.5200	0.5000

Table 6.6: AD Node Features Comparison Results

6.4.2 GNN Node Feature Comparison

After we have shown the availability of utilizing GNNs for accomplishing the Alzheimer's Disease classification, it is essential to test different node features to improve the performance. As shown in the table 6.6, our novel spatial node feature has reached state-of-the-art performance compared to other node features.

As shown in figure 6.8 and figure 6.9, it demonstrates that taking 1 as the node feature always performed the lowest accuracy since a homogeneous graph does not demonstrate any critical node and edge information. Most of the node features, such as the node strength and node degree, have shown inadequate robustness because they performed well on some classification models but performed poorly on others. For our novel spatial node feature, the nearly horizontal line chart clearly demonstrates its robustness. Nevertheless, it also performs best for most of the GNNs.

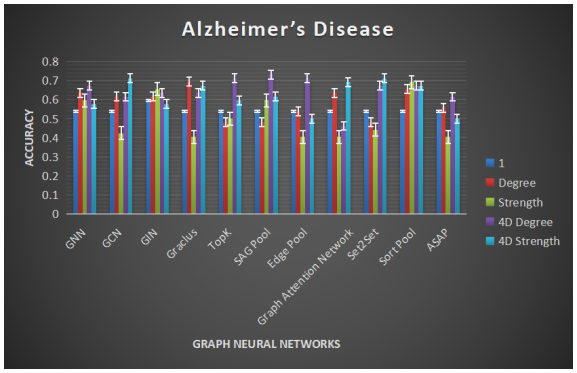

Figure 6.8: Histogram of AD Node Feature Comparison Results

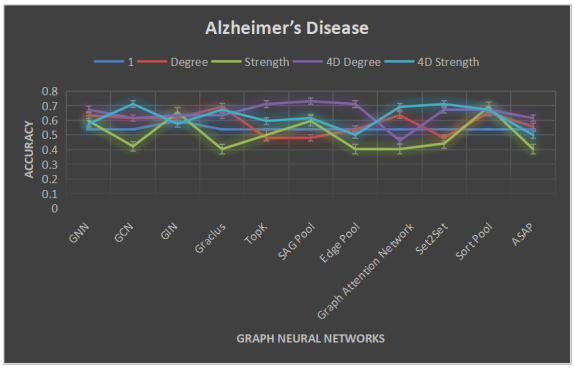

Figure 6.9: Line Chart of AD Node Feature Comparison Results

6.4.3 GNN vs Spatial

As mentioned before, graph neural networks are divided into spectral GNN and spatial GCN. In this case, it is necessary to compare them on the Alzheimer's Disease classification task. It took the spatial node feature as the input for all spectral neural networks and compared the results between spectral GNN and spatial GCN, as shown in figure **??**. It shows that our work performs better on many classification methods since it combines both topological features and spatial features.

6.4.4 Summary

In summary, our novel spatial node feature performs excellently on the AD datasets. Although it may not reach over 90% accuracy, it also reaches state-of-the-art performance. Moreover, it also clearly demonstrates its robustness compared to other node centrality features such as no degree. Thus, these results prove that our expectations, which aggregate both structural and spatial information, have been achieved.

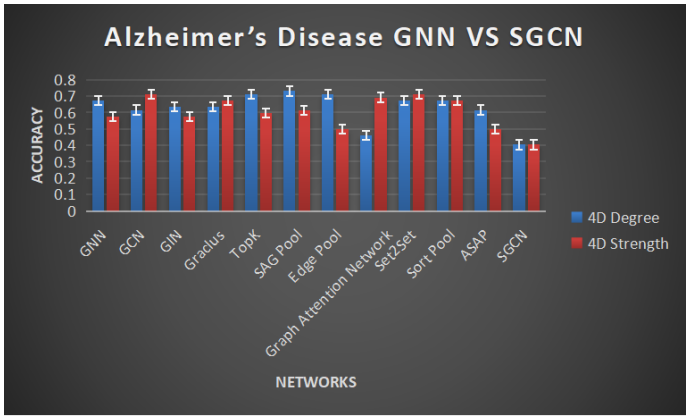

Figure 6.10: Comparison between CNN and GNN

6.5 Graph Neural Network GAN

After we had proved that GNNs are capable of accomplishing the Alzheimer's Disease classification task, we utilized the graph neural graph networks as the discriminator and classifier of the generative adversarial network. The discriminator is used to evaluate the distance between real and fake data. The classifier is used to identify whether the generated data belongs to Alzheimer's disease patients or ordinary people. Furthermore, we set the classical multilayer perceptron (MLP) as the generator, as shown in the figure 6.11. The table 6.7 shows the comparison of our graph neural network GAN and other traditional methods.[LWC21]. However, our work with novel spatial node features does not perform well compared to other methods. As shown in figure 6.13, the upper images are the actual data, and the lower images are the generated data. The images in the first and third columns are ordinary people's brain data. The images in the second column show the brain images of Alzheimer's Disease patients. As shown in figure 6.14 and figure 6.15, we also visualise the graph form of the diffusion brain MRI. These images clearly show that graph neural network GAN may not be sufficient for generating graph data. Moreover, the training process of the graph neural network GAN also demonstrates several issues. The first issue is that the discriminator does not optimise during the training process. The loss of the generator also does not perform a smooth line.

In our opinion, the results show the availability of graph neural network GAN for generating brain network data. However, the results also state that graph neural network GAN cannot effectively capture the topological and structural features of the brain data. We assume that the main issue happens at the generator. The generator we have utilized is a simple multilayer perceptron (MLP), and the input is Gaussian noise. These generators are implemented for image data which may not be sufficient for graph data. Besides that, graph neural networks consistently apply pooling layers which may cause the loss of information.

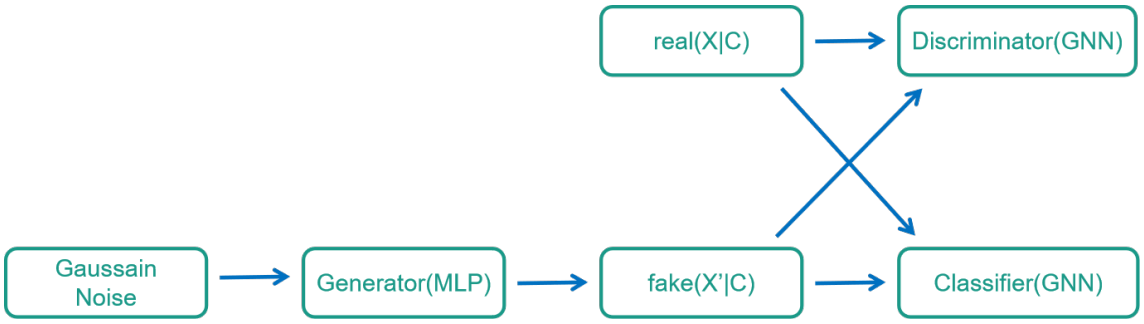

Figure 6.11: Workflow of Graph Neural Network GAN

	Kullback-Leibler Divergence		Maximum Mean Discrepancy	
	CN	AD	CN	AD
SMOTE	0.510	0.482	0.101	0.119
ADASYN	0.523	0.533	0.120	0.132
BrainNetGAN	0.473	0.460	0.110	0.123
GNN GAN	3.622	3.633	1.5807	1.5782

Table 6.7: Graph Neural Network GAN Results

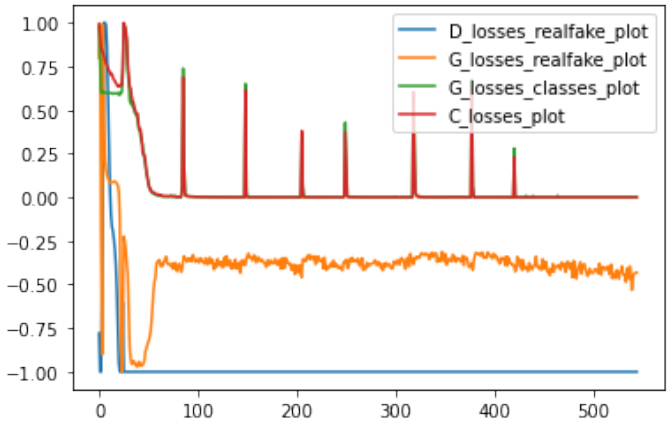

Figure 6.12: Training Process of Graph Neural Network GAN

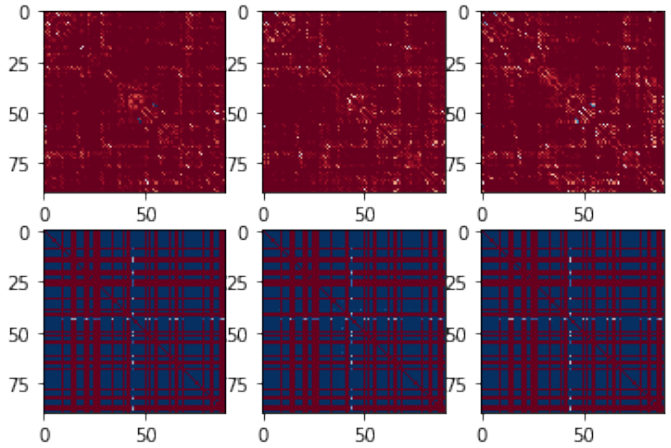

Figure 6.13: Generated Samples of Graph Neural Network GAN

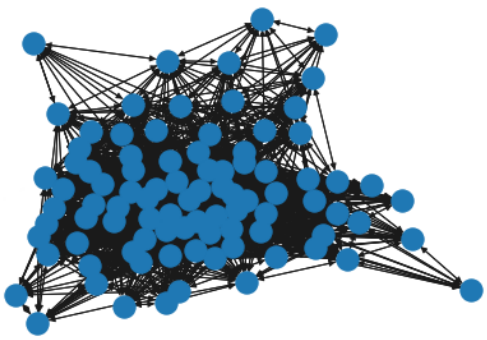

Figure 6.14: Real Brain Graph

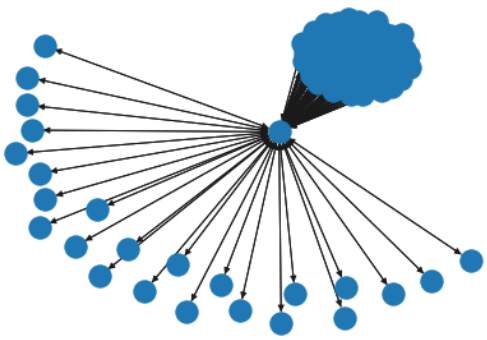

Figure 6.15: Generated Brain Graph

Chapter 7

Conclusions and Future Work

7.1 Conclusion

The goal of this research was to explore the possibilities of utilizing GNNs to improve the diagnosis and treatment of Alzheimer's Disease. To achieve this goal, we have utilized several different GNNs with our novel spatial node features to accomplish the classification task. Then we also presented the performance of graph neural network GAN for Alzheimer's disease data. From our exploration, we identify a successful spatial node feature that combines the spatial and topological information to reach state-of-the-art performance. According to the results, our method has successfully made a 2% improvement compared to the traditional methods. However, utilizing the spatial node feature and GNNs as the novel discriminator and classifier does not improve the performance of the data augmentation task. We believe that the questions happened in two folds, generator, and pooling layers.

In the process of discovering our final proposed method, we explored several alternative approaches comparing performance to CNN baselines. We demonstrated multiple existing classification spectral GNNs with several different node features. However, none of them have outperformed compared to the traditional CNN method. From our exploration and results, we argue that we progress and improve over existing successful approaches to classifying Alzheimer's Disease. Moreover, we demonstrate the potential for a GAN architecture with GNN participation for brain data augmentation tasks. I believe that our research is helpful for both diagnosis of Alzheimer's Disease and the augmentation of brain data. Therefore through this research, we are more to both bringing a successful complete AI-assisted treatment, which in theory should improve the whole area related to Alzheimer's Disease.

7.2 Future Work

There are several avenues of progression for this research that can be divided into two main categories: applying graph embedding methods on graph neural network GAN and implementing a total graph generator.

The main limitation of our exploration was caused by the handed-design node feature. It is quite a big flaw because we will never know whether there exist other hand-designed node features that can achieve better performance. In this case, the easiest way is to let the graph neural network design the node features automatically. Therefore, the graph embedding method will be the best way to encode the graphs with the most essential information. [Xu21] Otherwise, the main reason that Graph neural network GAN does not increase the performance of the data augmentation tasks is that the generator is a simple multilayer perceptron (MLP). Taking the gaussian noise as the generator's input may not be sufficient for graph datasets because gaussian noise does not contain any topological features. In this case, it is necessary to implement a novel graph generator.

Appendix A

First Appendix

Bibliography

[Ala18] Saleh Alamri. Pathways towards and away from alzheimers disease. *Health Care : Current Reviews*, 06, 01 2018.

[AW18] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? 05 2018.

[BBV07] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. The architecture of complex weighted networks. *Large Scale Structure and Dynamics of Complex Networks*, pages 67–92, 01 2007.

[BGA20] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. 11 2020.

[Cit10] Martin Citron. Alzheimer's disease: Strategies for disease modification. *Nature reviews. Drug discovery*, 9:387–98, 05 2010.

[Csi75] Imre Csiszar. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3, 02 1975.

[DD19] Michael DeTure and Dennis Dickson. The neuropathological diagnosis of alzheimer's disease. *Molecular Neurodegeneration*, 14, 12 2019.

[DGK05] Inderjit Dhillon, Yuqiang Guan, and Brian Kulis. A fast kernel-based multilevel algorithm for graph clustering. *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 629–634, 01 2005.

[DGK07] Inderjit Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29:1944–57, 12 2007.

[Die19] Frederik Diehl. Edge contraction pooling for graph neural networks. 05 2019.

[DST+20] Tomasz Danel, Przemysław Spurek, Jacek Tabor, Marek Śmieja, Łukasz Struski, Agnieszka Słowik, and Łukasz Maziarka. *Spatial Graph Convolutional Networks*, pages 668–675. 11 2020.

[EXXZ19] Ieee Engineering, Vol Xx, N Xx, and Yuan Zhang. Neuroimaging and machine learning for dementia diagnosis: Recent advancements and future prospects. *IEEE Reviews in Biomedical Engineering*, 12:19–33, 02 2019.

[FL19] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[Fre77] Linton Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 03 1977.

[FZB16] A. Fornito, A. Zalesky, and Edward Bullmore. *Fundamentals of Brain Network Analysis*. 03 2016.

[Gan18] Graham Ganssle. Expero blog | node classification by graph convolutional network, Jan 2018.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[GBR+12] A Gretton, K. Borgwardt, Malte Rasch, Bernhard Schölkopf, and AJ Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13:723–773, 03 2012.

[GBTH13] Alessandra Griffa, Philipp Baumann, Jean-Philippe Thiran, and Patric Hagmann. Structural connectomics in brain diseases. *NeuroImage*, 80, 04 2013.

[GJ79] Michael Garey and David Johnson. *Computer and Intractability: A Guide to the Theory of NP-Completeness*. 01 1979.

[GJ21] Hongyang Gao and Shuiwang Ji. Graph u-nets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 05 2021.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial nets. *ArXiv*, 06 2014.

[GSR+17] Justin Gilmer, Samuel Schoenholz, Patrick Riley, Oriol Vinyals, and George Dahl. Neural message passing for quantum chemistry. 04 2017.

[HCG+08] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher Honey, Van Wedeen, and Olaf Sporns. Mapping the structural core of human cerebral cortex. *PLoS biology*, 6:e159, 08 2008.

[KBM+16] Jeremy Kawahara, Colin Brown, Steven Miller, Brian Booth, Vann Chau, Ruth Grunau, Jill Zwicker, and Ghassan Hamarneh. Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *NeuroImage*, 146, 09 2016.

[KW16] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. 09 2016.

[LK21] Ramesh Lama and Goo-Rak Kwon. Diagnosis of alzheimer's disease using brain network. *Frontiners in Neuroscience*, 15, 02 2021.

[LLK19] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. 06 2019.

[LWC21] Chao Li, Yiran Wei, and Xi Chen. Brainnetgan: Data augmentation of brain connectivity using generative adversarial network for dementia classification. 03 2021.

[LZD+21] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence Staib, Pamela Ventola, and James Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 74:102233, 09 2021.

[MNG+22] Susanne Meinert, Nico Nowack, Dominik Grotegerd, Jonathan Repple, Nils Winter, Isabel Abheiden, Verena Enneking, Hannah Lemke, Lena Waltemate, Frederike Stein, Katharina Brosch, Simon Schmitt, Tina Meller, Julia-Katharina Pfarr, Kai Ringwald, Olaf Steinsträter, Marius Gruber, Igor Nenadić, Axel Krug, and Udo Dannlowski. Association of brain white matter microstructure with cognitive performance in major depressive disorder and healthy controls: a diffusion-tensor imaging study. *Molecular Psychiatry*, 27:1–8, 02 2022.

[RST20] Ekagra Ranjan, Soumya Sanyal, and Partha Talukdar. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:5470–5477, 04 2020.

[SGT+09] Franco Scarselli, Marco Gori, Ah Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 20:61–80, 01 2009.

[SLRPW21] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 2021. https://distill.pub/2021/gnn-intro.

[SMR+08] Kaustubh Supekar, Vinod Menon, Daniel Rubin, Mark Musen, and Michael Greicius. Network analysis of intrinsic functional brain connectivity in alzheimer's disease. *PLoS computational biology*, 4:e1000100, 06 2008.

[SVR+21] Markus Schirmer, Archana Venkataraman, Islem Rekik, Minjeong Kim, Stewart Mostofsky, Mary Beth Nebel, Keri Rosch, Karen Seymour, Deana Crocetti, Hassna Irzan, Michael Hütel, Sebastien Ourselin, Neil Marlow, Andrew Melbourne, Egor Levchenko, Shuo Zhou, Mwiza Kunda, Haiping Lu, Nicha Dvornek, and Ai Wern Chung. Neuropsychiatric disease classification using functional connectomics - results of the connectomics in neuroimaging transfer learning challenge. *Medical Image Analysis*, 70:101972, 01 2021.

[VBK16] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. 11 2016.

[VNL+20] Maria Valueva, Nikolay Nagornov, Pavel Lyakhov, G.V. Valuev, and N.I. Chervyakov. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 177, 05 2020.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 06 2017.

[Wen06] Gary Wenk. Neuropathologic changes in alzheimer's disease: Potential targets for treatment. *The Journal of clinical psychiatry*, 67 Suppl 3:3–7; quiz 23, 02 2006.

[WLP21] Yiran Wei, Chao Li, and Stephen Price. Quantifying structural connectivity in brain tumor patients. pages 519–529, 09 2021.

[WZY17] Shiguang Wang, Lili Zheng, and Dexin Yu. node degree, 01 2017.

[XHLJ18] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? 10 2018.

[Xu21] Mengjia Xu. Understanding graph embedding methods and their applications. *SIAM Review*, 63:825–853, 01 2021.

[ZCNC18] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32, 04 2018.

[ZMJ14] Mohammed Zaki and Wagner Meira Jr. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. 05 2014.