Classification of Developmental and Brain Disorders via Graph Convolutional Aggregation

Ibrahim Salim and A. Ben Hamza
Concordia Institute for Information Systems Engineering
Concordia University, Montreal, QC, Canada

Abstract

While graph convolution based methods have become the defacto standard for graph representation learning, their applications to disease prediction tasks remain quite limited, particularly in the classification of neurodevelopmental and neurodegenerative brain disorders. In this paper, we introduce an aggregator normalization graph convolutional network by leveraging aggregation in graph sampling, as well as skip connections and identity mapping. The proposed model learns discriminative graph node representations by incorporating both imaging and non-imaging features into the graph nodes and edges, respectively, with the aim to augment predictive capabilities. We benchmark our model against several recent baseline methods on two large datasets, ABIDE and ADNI, for the prediction of autism spectrum disorder and Alzheimer's disease, respectively. Experimental results demonstrate the competitive performance of our approach in comparison with recent baselines in terms of several evaluation metrics, achieving relative improvements of 50% and 13.56% in classification accuracy over graph convolutional networks on ABIDE and ADNI, respectively.

Keywords: Disease prediction; graph convolution; skip connection; aggregation; autism spectrum disorder; Alzheimer's disease.

1 Introduction

Understanding how the brain develops is vital to designing prediction models and formulating treatments for a variety of developmental disorders and degenerative neurological diseases such as autism spectrum disorder and Alzheimer's disease, which are devastating illnesses that have touched the lives of millions of families around the world, not only in personal anguish, but also in soaring healthcare costs [1]. Autism spectrum disorder is a neurodevelopmental disability that affects how a person communicates, learns and socializes with others, whereas Alzheimer's disease is a chronic neurodegenerative brain disorder that slowly destroys brain cells, causing memory loss and cognitive decline over time.

Graph-structured data is ubiquitous in a wide range of real-world application domains, including social networks, biological protein-protein interaction networks, molecular graph structures, and brain connectivity networks. Graphs provide a flexible way to inherently represent real-world entities as a set of nodes and their interactions as a set of edges. A case in point: for brain analysis in populations and diagnosis, we model populations as graphs, where each node represents a subject with an associated node feature vector obtained from imaging data, and each edge represents a pairwise similarity between two subjects with an edge feature vector acquired from non-imaging data.

In recent years, there has been a surge of interest in extending deep learning approaches to non-Euclidean domains thanks, in large part, to the prevalence and increasing proliferation of graph-structured data [2–5]. Advances in deep learning have spawned significant efforts to facilitate, for instance, the clinical diagnosis of brain diseases. Graph convolutional networks (GCNs), which generalize convolutional neural networks to graph-structured data by leveraging spectral graph theory and its extensions, have gained popularity in graph representation learning [2] for their ability to capture the graph structure. GCN uses a layer-wise propagation rule based on a first-order approximation of spectral graph convolutions, where the feature vector of each graph node is updated by applying a weighted sum of the features of its immediate neighboring nodes. Wu *et al.* [3] introduce a simple graph convolution by removing the nonlinear transition functions between the layers of graph convolutional networks and collapsing the resulting function into a single linear transformation via the powers of the normalized adjacency matrix with added self-loops for all graph nodes. However, this simple graph convolution acts as a low-pass filter, which attenuates all but the zero frequency, causing oversmoothing. Zeng *et al.* [4] propose a graph sampling based learning method by sampling the training graph in lieu of nodes or edges across GCN layers, as well as eliminating biases in minibatch estimation via aggregator normalization techniques. Chen *et al.* [5] present an extension of the GCN model using skip connections from the input layer and identity mapping with the learnable weight matrix of each layer in a bid to alleviate the oversmoothing problem, which is a common effect of increasing the network depth.

The primary objective of graph convolution based methods is to learn node representations that encode structural information about the graph. These learned node representations can then be used as input to machine learning models for downstream tasks such as node classification whose goal is to predict the most probable labels of nodes in a graph. For instance, in brain diagnosis tasks, which is the focus of this work, we want to classify subjects as diseased or healthy by predicting the node labels in a population graph. Graph convolution based methods have

recently become prevalent in the biomedical and medical imaging domains [6–10] due largely to the fact that neuroimaging provides valuable information about the diagnosis and progression of brain diseases. Built on top of graph signal processing approaches [11], GCNs have shown promising results in metric learning and classification tasks on brain connectivity networks [12,13]. Ktena *et al.* [12] propose to learn a graph similarity metric using a siamese graph convolutional neural network in a supervised fashion, yielding encouraging results in individual subject classification and manifold learning tasks. Similarly, Ma *et al.* [13] introduce a higher-order siamese graph convolutional neural network for multi-subject brain analysis in health and neuropsychiatric disorders by incorporating higher-order proximity in graph convolutions, with the goal of characterizing the community structure of brain connectivity networks and learning the similarity among magnetic resonance imaging (fMRI) brain connectivity networks extracted from multiple subjects. While GCNs have also been successfully used in the prediction of developmental and brain disorders such as autism spectrum disorder (ASD) and Alzheimer's disease (AD) [14–16], they are prone to the oversmoothing problem, where learned node representations become similar due to repeated graph convolutions as the network depth increases. In other words, when the number of GCN layers increases, the learned node representations tend to converge to indistinguishable feature vectors, resulting in performance degradation and less expressiveness; and hence the model becomes less aware of the graph structure.

In order to overcome the aforementioned issues, we propose an aggregator normalization graph convolutional network (ANGCN) with skip connections and identity mapping for the detection of neurodevelopmental and neurodegenerative brain disorders by integrating both imaging and non-imaging features into the graph nodes and edges, respectively. We formulate the disease prediction problem as a semi-supervised node classification on population graphs. The main contributions of this work can be summarized as follows:

- We propose a novel graph convolutional aggregation approach with skip connections and identity mapping for node classification by effectively integrating into the graph both imaging and non-imaging information.
- We employ an aggregator normalization mechanism for feature propagation in an effort to eliminate bias in mini-batch estimation.
- We show through experimental results that our model yields competitive performance in comparison with strong baselines on two large benchmark datasets.

The remainder of this paper is organized as follows. In Section 2, we review important relevant work. In Section 3, we present the problem formulation as a semi-supervised node classification task, and then we introduce a two-stage graph convolutional aggregation framework for disease prediction. In the first stage, we construct a population graph comprised of a node set and an edge set with complementary imaging and non-imaging data, respectively. Each graph node represents a subject with an associated feature vector extracted from imaging data, and each edge captures similarities between a pair of subjects with non-imaging data integrated into the edge weight. In the second stage, we design an aggregator normalization graph convolutional network architecture by leveraging skip connections, identity mapping and aggregation in graph sampling. In Section 4, we present experimental results to demonstrate the competitive performance of our approach in comparison with graph-based methods for brain disease prediction. Finally, we conclude in Section 5 and highlight some promising directions for future work.

2 Related Work

The basic objective of node classification in populations and diagnosis is to predict the most probable labels of nodes in a population graph, where each subject is represented by a node and each edge encodes the pairwise similarity between a pair of connected nodes. To achieve this objective, various graph convolution based methods have been proposed with the aim of distinguishing between diseased patients and healthy controls by predicting the node labels (i.e. clinical status of subjects). In semi-supervised node classification, the amount of labeled nodes for model training is typically small and the goal is to predict the labels of a large number of unlabeled nodes by learning a prediction rule from both labeled and unlabeled nodes in order to improve model performance.

Graph Convolutional Networks. GCNs have recently become the model of choice in semi-supervised node classification tasks [2]. GCN uses an efficient layer-wise propagation rule, which is based on a first-order approximation of spectral graph convolutions. The feature vector of each graph node is updated by essentially applying a weighted sum of the features of its neighboring nodes. Xu *et al.* [17] introduce a graph wavelet neural network, which is a GCN-based architecture that uses spectral graph wavelets in lieu of graph Fourier bases to define a graph convolution. Despite the fact that spectral graph wavelets can yield localization of graph signals in both spatial and spectral domains, they require explicit computation of the Laplacian eigenbasis, leading to a high computational complexity, especially for large graphs.

While GCNs have shown great promise, achieving state-of-the-art performance in semi-supervised node classification tasks, they are prone to oversmoothing the node features. In fact, the neighborhood aggregation scheme (i.e. graph convolution) of GCN is tantamount to applying Laplacian smoothing [18], which replaces each graph node with the average of its immediate neighbors. Therefore, repeated application of GCN yields smoother and smoother versions of the initial node features as the number of the network's layers increases. As a result, the node features in deeper layers will eventually converge to the same value, and hence become too similar across different classes. Wu *et al.* [3] introduce a simple graph convolution by removing the nonlinear transition functions between the layers of graph convolutional networks and collapsing the resulting function into a single linear transformation via the powers of the normalized adjacency matrix with added self-loops for all graph

nodes. However, this simple graph convolution acts as a low-pass filter, which attenuates all but the zero frequency, causing oversmoothing. Significant strides have been made toward remedying the problem of oversmoothing in GCNs [5, 19, 20]. Xu *et al.* [19] propose jumping knowledge networks, which employ dense skip connections to connect each layer of the network with the last layer to preserve the locality of node representations in order circumvent oversmoothing. In [20], a normalization layer, which helps avoid oversmoothing by preventing learned representations of distant nodes from becoming indistinguishable, has been proposed. This normalization layer is performed on intermediate layers during training, and the aim is to apply smoothing over nodes within the same cluster while avoiding smoothing over nodes from different clusters. Chen *et al.* [5] design a deep graph convolutional with initial residual and identity mapping to tackle the problem of oversmoothing by adding an identity matrix to the learnable weight matrix and skip connections from the initial feature matrix.

Disease Prediction. GCNs have recently shown great potential in neuroimaging and computer aided diagnosis, especially in the prediction of brain diseases such as autism spectrum disorder and Alzheimer's disease [14–16, 21–24]. Using a graph convolutional neural network model consisting of a fully convolutional GCN with several hidden layers activated via the Rectified Linear Unit function, Parisot *et al.* [14] introduce a disease prediction framework. It involves modeling a population as a graph with nodes representing subjects and edges encoding the similarity between a pair of subjects by combining imaging and non-imaging information in order to improve model classification performance with the goal of distinguishing between patients with autism spectrum disorder and healthy controls, as well as predicting whether a patient with mild cognitive impairment will convert to Alzheimer's disease. The graph nodes are associated with imaging-based features, while non-imaging data is integrated into the edge weights. To learn an adaptive graph representation for GCN learning, Zheng *et al.* [15] integrate graph learning and graph convolution to develop an end-to-end multimodal graph learning approach for disease prediction via a multi-modal fusion module, which fuses the features of each modality by leveraging the correlation and complementarity between the modalities. Cao *et al.* [16] introduce a deep learning model using a multi-layer GCN in conjunction with residual neural networks to tackle the vanishing gradient problem, and the DropEdge technique [25] to alleviate overfitting and oversmoothing, which are two major challenges in developing deep GCNs for node classification. Similar to Dropout technique that randomly sets the outgoing edges of hidden units to zero at each update of the training phase, DropEdge can be regarded as an extension of Dropout to graph edges. Inspired by the Inception network in convolutional neural networks, Kazi *et al.* [21] propose an Inception graph convolutional network for disease prediction tasks with complementary imaging and non-imaging multi-modal data by leveraging spectral convolutions with different kernel sizes, showing improved performance over regular GCN architectures. Cosmo *et al.* [22] present an end-to-end trainable graph learning architecture for dynamic and localized graph pruning with the aim to build a node classification model consisting of few graph convolutional layers, followed by a fully connected layer to predict the patient label. Building upon GCNs, Jiang *et al.* [26] introduce a hierarchical GCN model for graph embedding learning of brain network and brain disorders prediction by hierarchically learning deep representations from functional fMRI brain connectivity networks in order to improve classification performance for disease diagnosis. Pan *et al.* [23] propose a diagnosis classification framework that incorporates self-attention graph pooling and graph convolutional networks by extracting features from the non-Euclidean brain network, as well as fusing both imaging and non-imaging information with the aim to detect inter-group heterogeneity and intra-group homogeneity regarding brain activities. While these approaches have shown promising results in brain disease prediction tasks, they are, however, prone to the issue of oversmoothing.

3 Method

In this section, we introduce our notation and formulate the disease prediction problem as a semi-supervised node classification task on population graphs, which are used to model pairwise relations (edges) between subjects (nodes). Each graph node and edge weight are associated with complementary imaging and non-imaging data, respectively. Then, we present the main building blocks of the proposed network architecture for graph representation learning and semi-supervised node classification.

3.1 Preliminaries and Problem Statement

Basic Notions. Let $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ be a graph, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of N nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. We denote by $\mathbf{A} = (\mathbf{A}_{ij})$ an $N \times N$ adjacency matrix (binary or real-valued) whose (i, j)-th entry $\mathbf{A}_{ij}$ is equal to the weight of the edge between neighboring nodes i and j, and 0 otherwise. We also denote by $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)^\intercal$ an $N \times F$ feature matrix of node attributes, where $\mathbf{x}_i$ is an F-dimensional row vector for node i.

Learning latent representations of nodes in a graph aims at encoding the graph structure into low-dimensional embeddings, such that both structural and semantic information are captured. More precisely, the purpose of network/graph embedding is to learn a mapping $\varphi : \mathcal{V} \rightarrow \mathbb{R}^P$ that maps each node i to a P-dimensional vector $\mathbf{z}_i$, where $P \ll N$. These learned node embeddings can then be used as input to learning algorithms for downstream tasks, such as node classification.

Problem Statement. Given the labels of a subset of the graph nodes (or their corresponding final output embeddings), the objective of semi-supervised learning is to predict the unknown labels of the other nodes. More specifically, let $\mathcal{D}_l = \{(\mathbf{z}_i, y_i)\}_{i=1}^{N_l}$ be the set of labeled final output node embeddings $\mathbf{z}_i \in \mathbb{R}^P$ with associated known labels $y_i \in \mathcal{Y}_l$, and $\mathcal{D}_u = \{\mathbf{z}_i\}_{i=N_l+1}^{N_l+N_u}$ be the set of unlabeled final output node embeddings, where $N_l + N_u = N$. Then, the problem of semi-supervised node classification is to learn a classifier $f : \mathcal{V} \rightarrow \mathcal{Y}_l$. That is, the goal is to predict the labels of the set $\mathcal{D}_u$.

It is important to note that for multi-class classification problems, the label of each node i (or its final output embedding $\mathbf{z}_i$) in the labeled set $\mathcal{D}_l$ can be represented as a C-dimensional one-hot vector $\mathbf{y}_i \in \{0,1\}^C$, where C is the number of classes with 0 and 1 representing "healthy" and "diseased" status of the subjects, respectively.

3.2 Proposed Model

We now describe our proposed model, a two-stage approach for graph representation learning and semi-supervised node classification. The aim is to learn discriminative node embeddings for computer aided diagnosis. In the first stage, we construct a population graph, which is a vital step in designing a GCN-based prediction model since GCNs rely on the affinity matrix between subjects to update their layer-wise feature propagation rules. Hence, to fully exploit the expressive power of GCNs, an appropriately constructed graph that accurately explains the similarity between subjects is of paramount importance in graph representation learning, especially in computer aided diagnosis tasks. In the second stage, we introduce a disease prediction model by leveraging graph convolutional aggregation in conjunction with skip connections and identity mapping.

3.2.1 Population Graph Construction

Following the population graph construction in graph convolutional networks for disease prediction [14], we also combine both imaging and non-imaging data in our proposed approach. More specifically, we model a population as a graph consisting of nodes representing subjects and edges capturing pairwise similarities between subjects. Each node has a feature vector extracted from imaging data, whereas each edge weight represents phenotypic (i.e. non-imaging) data. The graph construction is shown in Figure 1, where the nodes are associated with imaging-based feature vectors, while phenotypic (non-imaging) information is incorporated as edge weights.

Let $\{M_1, \ldots, M_T\}$ be a set of T non-imaging phenotypic measures such as a subject's age or gender. The adjacency matrix $\mathbf{A} = (\mathbf{A}_{ij})$ of a population graph comprised of N subjects is defined as

$$\mathbf{A}_{ij} = K(i,j) \sum_{t=1}^{T} d(M_t(i), M_t(j)), \tag{1}$$

where $K(i,j) = \text{similarity}(S_i, S_j)$ denotes a kernel similarity between subjects S_i and S_j (i.e. edge weight between graph nodes i and j), and d is a pairwise distance between phenotypic measures. The kernel similarity measure $K(i,j)$ is given by

$$K(i,j) = \exp\Big(-\frac{\rho(\mathbf{x}_i, \mathbf{x}_j)^2}{2\sigma^2}\Big), \tag{2}$$

where σ is a smoothing parameter, which determines the width of the kernel, and ρ is the correlation distance between feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ for nodes i and j, respectively,

$$\rho(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{(\mathbf{x}_i - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_j)^\intercal}{\|\mathbf{x}_i - \bar{\mathbf{x}}_i\| \|\mathbf{x}_j - \bar{\mathbf{x}}_j\|}, \tag{3}$$

with $\bar{\mathbf{x}}_i = (\mathbf{x}_i \mathbf{1}/N)\mathbf{1}^\intercal$ and $\bar{\mathbf{x}}_j = (\mathbf{x}_j \mathbf{1}/N)\mathbf{1}^\intercal$ denoting row vectors whose elements are all equal to the mean of the components of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, and $\mathbf{1}$ is an N-dimensional column vector of all ones.

The pairwise distance between phenotypic measures is defined depending on the kind of phenotypic data incorporated in the graph. Most phenotypic data can be classified into two main categories: qualitative (e.g. subject's gender) and quantitative (e.g. subject's age). For qualitative data, the distance measure is defined as

$$d(M_t(i), M_t(j)) = \begin{cases} 1 & \text{if } M_t(i) = M_t(j) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

and for quantitative data, it is given by

$$d(M_t(i), M_t(j)) = \begin{cases} 1 & \text{if } |M_t(i) - M_t(j)| < \tau, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

where τ is a given threshold.

3.2.2 Disease Prediction Model

Graph convolutional networks learn a new feature representation for each node such that nodes with the same labels have similar features [2].

Feature Diffusion. We denote by $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ the adjacency matrix with self-added loops, where $\mathbf{I}$ is the identity matrix. The layer-wise feature diffusion rule of an L-layer GCN is given by

$$\mathbf{S}^{(\ell)} = \hat{\mathbf{A}}\mathbf{H}^{(\ell)}, \quad \ell = 0, \ldots, L-1, \tag{6}$$

where $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ is the normalized adjacency matrix with self-added loops, $\tilde{\mathbf{D}} = \text{diag}(\tilde{\mathbf{A}}\mathbf{1})$ is the diagonal degree matrix, and $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N \times F_\ell}$ is the input feature matrix of the ℓ-th layer with F_ℓ feature maps. The input of the first layer is the original feature matrix $\mathbf{H}^{(0)} = \mathbf{X}$.

Aggregated Feature Diffusion. Inspired by the aggregation mechanism in graph sampling [4], we define a layer-wise aggregated feature diffusion rule for node features in the ℓ-th layer as follows:

$$\mathbf{S}^{(\ell)} = (\hat{\mathbf{A}} \odot \mathbf{\Gamma})\mathbf{H}^{(\ell)}, \tag{7}$$

where $\odot$ denote element-wise matrix multiplication, and $\mathbf{\Gamma} = (\gamma_{ij})$ is an $N \times N$ aggregation matrix. Each element γ_{ij} is an aggregator normalization constant given by

$$\gamma_{ij} = \frac{C_i}{C_{ij}}, \tag{8}$$

where C_i and C_{ij} denote the number of times the node $i \in \mathcal{V}$ or edge $(i,j) \in \mathcal{E}$ appears in the subgraphs of $\mathbb{G} = (\mathcal{V}, \mathcal{E})$, respectively. These subgraphs are obtained by running the GraphSaint sampler repeatedly before the training starts [4].

Learning Node Embeddings. Motivated by the good performance of graph sampling and identity mapping in alleviating the oversmoothing problem in graph representation learning [3–5,27–29], we propose an aggregator normalization graph convolutional network (AN-GCN) by leveraging aggregation in

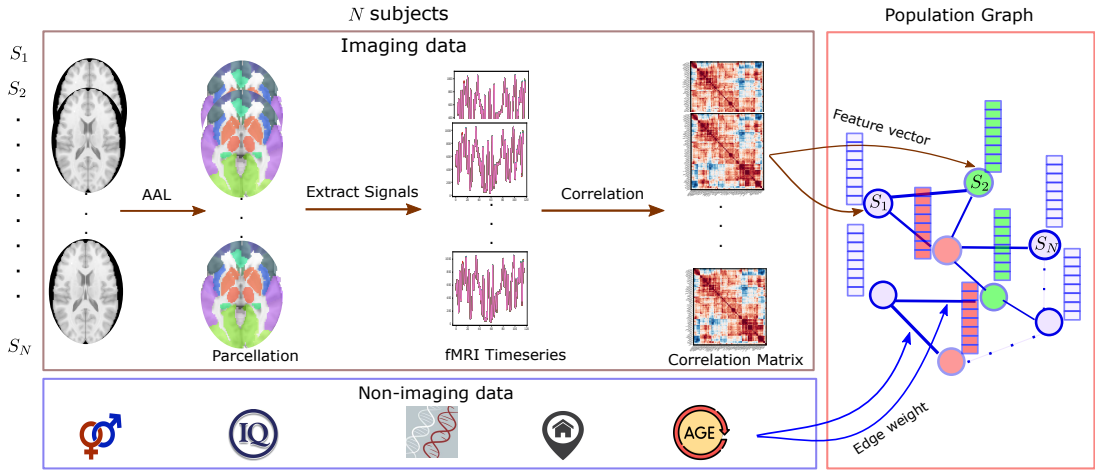

Figure 1: Graph construction from N subjects using imaging and non-imaging data. For imaging data, we employ Automated Anatomical Labeling (AAL) to perform brain parcellation.

graph sampling, as well as skip connections and identity mapping. The output feature matrix $\mathbf{H}^{(\ell+1)}$ of our proposed AN-GCN model is obtained by applying the following layer-wise propagation rule

$$\begin{aligned}\mathbf{H}^{(\ell+1)} = \sigma\Big(&\big(1-\alpha_\ell\big)(\hat{\mathbf{A}}\odot\mathbf{\Gamma})\mathbf{H}^{(\ell)} \\ &+ \beta_\ell(\hat{\mathbf{A}}\odot\mathbf{\Gamma})\mathbf{H}^{(\ell)}\big(\mathbf{I}+\mathbf{W}^{(\ell)}\big) \\ &+ \alpha_\ell\mathbf{X} + \beta_\ell\mathbf{X}\big(\mathbf{I}+\mathbf{W}^{(\ell)}\big)\Big),\end{aligned} \tag{9}$$

where α_ℓ and β_ℓ are nonnegative hyper-parameters in the interval $[0,1]$ and are often fine-tuned via grid search, and $\sigma(.)$ is a point-wise non-linear activation function such as $\text{ReLU}(.) = \max(0,.)$. The use of skip connections ensures that the final representation of each node retains at least a percentage α_ℓ of feature data from the input layer, while identity mapping not only imposes regularization on the learnable weight matrix in order to avoid over-fitting, but it is also beneficial to semi-supervised learning tasks where training data is limited [30].

Model Prediction. The embedding matrix $\mathbf{Z} = \mathbf{H}^{(L)}$ of the last layer of AN-GCN contains the final output node embeddings, and captures the neighborhood structural information of the graph within L hops. This final node representation can be used as input for node classification. To this end, we apply a softmax classifier as follows:

$$\hat{\mathbf{Y}} = \text{softmax}(\mathbf{Z}), \tag{10}$$

where $\hat{\mathbf{Y}} \in \mathbb{R}^{N\times C}$ is the matrix of predicted labels for graph nodes, and C is the total number of classes. The softmax classifier is a generalization of the binary logistic regression classifier to multiple classes, and as the name suggests it uses the softmax function that turns a vector of C real-valued class scores into a vector of C normalized positive scores that sum to 1. In other words, the softmax classifier returns probability scores for all classes.

Model Training. For semi-supervised multi-class classification, the neural network weight parameters are learned by minimizing the cross-entropy loss function

$$\mathcal{L} = -\sum_{i\in\mathcal{Y}_l}\sum_{c=1}^{C}\mathbf{Y}_{ic}\log\hat{\mathbf{Y}}_{ic}, \tag{11}$$

over the set $\mathcal{Y}_l$ of all labeled nodes, where $\mathbf{Y}_{ic}$ is equal 1 if node i belongs to class c, and 0 otherwise; and $\hat{\mathbf{Y}}_{ic}$ is the (i,c)-element of the matrix $\hat{\mathbf{Y}}$ from the softmax function, i.e. the probability that the network associates the i-th node with class c. During training, the network parameters are updated using the Adam optimizer [31].

4 Experiments

In this section, we conduct several experiments to assess the performance of the proposed AN-GCN model on two standard datasets for disease prediction. More specifically, we address the disease prediction problem as a semi-supervised node classification task, and the goal is predict the label (i.e. clinical status) of a test node (i.e. subject) in a population graph as diseased or healthy, where only a small number of nodes are labeled. The effectiveness of our model is validated through experimental comparison with strong baseline methods. While presenting and analyzing our experimental results, we aim to answer the following main research questions (RQs):

- **RQ1:** How does AN-GCN perform in comparison with state-of-the-art disease prediction models?
- **RQ2:** How does AN-GCN alleviate the oversmoothing problem?
- **RQ3:** What is the effect of hyperparameters on the performance of AN-GCN?

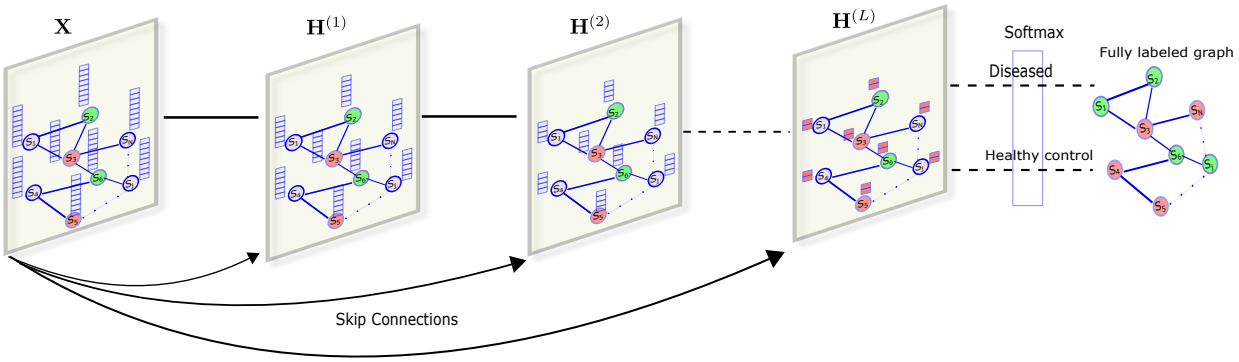

Figure 2: Schematic layout of the proposed AN-GCN architecture.

4.1 Experimental Setup

4.1.1 Datasets

We evaluate the proposed model on two large datasets, namely ABIDE and ADNI.

- **ABIDE Dataset:** The Autism Brain Imaging Data Exchange (ABIDE)[1] initiative aggregates resting-state functional magnetic resonance imaging (rs-fMRI) and phenotypic data of 1112 subjects from various international brain imaging laboratories. We select a set of 871 subjects, consisting of 403 ASD patients and 468 healthy controls (HCs). As a result of the different acquisition sites, the ABIDE dataset is heterogeneous, and the aim is to separate ASD subjects from healthy controls.
- **ADNI Dataset:** The Alzheimer's Disease Neuroimaging Initiative (ADNI)[2] is a North American multisite study designed to develop clinical, neuroimaging techniques, biochemical and genetic biomarkers for the early detection and tracking of patients with Alzheimer's disease (AD), as well as subjects with mild AD, normal subjects, and subjects with mild cognitive impairment (MCI). ADNI has recruited more than 1700 adults, aged 55 to 90 years, from over 50 sites across the United States and Canada for its four studies (ADNI-1, 2, 3 and -GO). We select a set of 573 participants, comprised of 402 HC individuals and 171 MCI subjects. The aim is to predict whether an MCI subject will convert to AD.

4.1.2 Data Preprocessing

For fair comparison, we follow the same data preprocessing procedure laid out in the GCN baseline [14]. For preprocessing of the ABIDE dataset, we use the Configurable Pipeline for the Analysis of Connectomes (C-PAC), which includes skull stripping, slice timing correction, motion correction, global mean intensity normalization, nuisance signal regression, band-pass filtering (0.01-0.1Hz), and registration of fMRI images to a standard anatomical space. Then, the mean timeseries for a set of cortical and subcortical regions of interest (ROIs) extracted from the Harvard Oxford atlas are computed and standardized using z-score normalization to ensure the timeseries distributions have mean zero mean and unit variance. The goal of z-score normalization is to transform timeseries to be on a similar scale in an effort to improve the performance and training stability of the model. Subsequently, we compute N connectivity matrices using the Pearson's correlation coefficient between the representative rs-fMRI timeseries of each ROI in the Harvard Oxford atlas. Since z-scores are not necessarily normally distributed, we apply Fisher z-transformation, which is the inverse hyperbolic tangent function that converts Pearson's correlation coefficient to a normally distributed variable. In other words, the correlation matrices are Fisher transformed in order to convert the skewed distribution of the correlation coefficient into a distribution that is approximately normal. It is also worth pointing out that the variance of the Fisher transformed distribution is independent of the correlation, whereas the variance of the sampling distribution of the correlation coefficient depends on the correlation. For the edge weights of the population graph on the ABIDE datset, we incorporate the subject's gender, age and acquisition site as as phenotypic measures. For the ADNI dataset, we parcellate each 3D brain volume into N ROIs using Automated Anatomical Labeling (AAL), followed by computing the connectivity matrices between timeseries. The edge weights of population graph on the ADNI dataset consist of the subject's gender and age as phenotypic measures. Since the correlation matrix is symmetric, it suffices to use either its upper or lower triangular part. Hence, we take the upper triangular part and vectorize it to obtain a feature vector whose dimension is then further reduced using recursive feature elimination via a ridge classifier.

4.1.3 Performance Evaluation Metrics

A classification model's performance is normally evaluated by applying it to test data with known target values and comparing the predicted values to the known values. We use Accuracy (Acc), Area Under Curve (AUC), Recall, Precision, F1 score, Matthews Correlation Coefficient (MCC), and Cohen's kappa (κ) as evaluation metrics, which are defined as

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$

[1] http://preprocessed-connectomes-project.org/abide/
[2] http://adni.loni.usc.edu/

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TN} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FN})}},$$

and

$$\kappa = \frac{2 \times (\text{TP} \times \text{TN} - \text{FP} \times \text{FN})}{(\text{TP} + \text{FP}) \times (\text{TN} + \text{FP}) + (\text{TP} + \text{FN}) \times (\text{TN} + \text{FN})},$$

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

The F1-score is defined as the harmonic mean of precision and recall. The harmonic mean is more intuitive than the arithmetic mean when computing a mean of ratios. The F1-score will only be high if both precision and recall have high values. This is due to the fact that the harmonic mean of two numbers is always closer to their minimum.

We also use AUC, the area under the receiving operating characteristic (ROC) curve, as a metric. AUC summarizes the information contained in the ROC curve, which plots the true positive rate versus the false positive rate, at various thresholds. Larger AUC values indicate better performance at distinguishing between healthy and diseased subjects. An uninformative classifier has an AUC equal to 50% or less. An AUC of 50% corresponds to a random classifier (i.e. for every correct prediction, the next prediction will be incorrect), whereas an AUC score smaller than 50% indicates that the classifier performs worse than a random one.

4.1.4 Baseline Methods

We evaluate the performance of the proposed AN-DGCN model against various graph convolutional based methods for computer aided diagnosis, including GCN for disease prediction [14], multi-modal graph learning (MMGL) for disease prediction [15], DeepGCN for autism spectrum disorder identification from multi-site resting-state data [16], InceptionGCN for disease prediction [21], latent-graph learning (LGL) for disease prediction [22], edge-variational graph convolutional network (EV-GCN) for uncertainty-aware disease prediction [32], down-sampling and multi-modal learning (DS-MML) for identifying autism spectrum disorder [23], hierarchical graph convolution network (HI-GCN) for brain disorders prediction [26], brain connectivity via graph convolution network (BN-GCN) for Alzheimer's Disease [33], and mutual multi-scale triplet graph convolutional network (MMTGCN) for brain disorders classification [24]. We also compare our model to logistic regression, gradient boosting, and tensor-train, high-order pooling and semi-supervised learning-based generative adversarial network (THS-GAN) [34].

4.1.5 Implementation Details

All experiments are carried out on a Linux workstation running Intel(R) CPU 2.40 GHz and 128-GB RAM with an V100-SXM2 16-GB GPU. The proposed model is implemented in PyTorch and trained for 150 and 100 epochs on the ABIDE and ADNI datasets, respectively, using Adam optimizer with a learning rate of 10^{-3}. The values of the hyperparameters α_ℓ and β_ℓ are set to 0.1 and 0.3 for the ABIDE dataset, and 0.1 and 0.2 for the ADNI dataset, respectively, via grid search with cross-validation on the training set. We use a stratified 10-fold cross-validation strategy. We also set the number of layers for our model to $L = 10$. The training is stopped when the validation loss does not decrease after 10 consecutive epochs. The values of the cross-entropy metric are recorded at the end of each epoch on the training set. The performance comparison plots of AN-GCN and GCN over training epochs on the training set of the ABIDE dataset are visualized in Figure 3, which shows that both models yield comparable training loss values. However, as the number of epochs increases, our model yields lower training loss values, indicating better predictive accuracy.

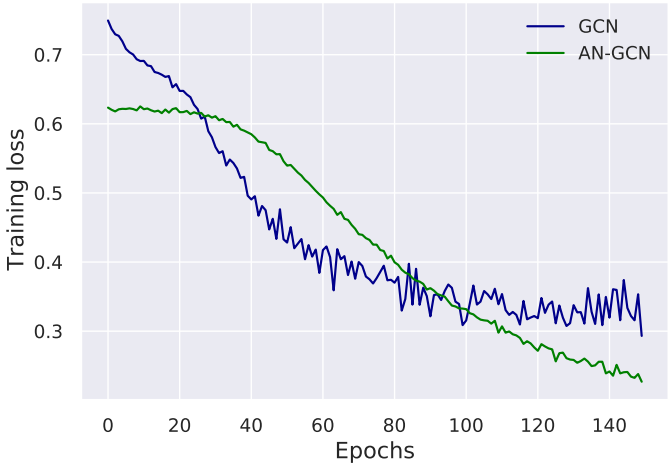

Figure 3: Model training history comparison between GCN and our AN-GCN model on the ABIDE dataset.

4.2 Experimental Results and Analysis

In order to answer **RQ1**, we report the classification performance of AN-GCN and baseline methods in Table 1 using seven evaluation metrics, including average accuracy, AUC and F1-score. Each metric is averaged across all test samples. As can be seen, the results show that our model outperforms all the baseline methods on the ABIDE dataset, achieving relative improvements of 50.33%, 34.10%, 50.33%, 12.25% and 50.33% over GCN in terms of accuracy, AUC, F1-score, recall and precision, respectively. The relative improvements over GCN are significant in terms of κ and MCC. Compared to the strongest baseline, our model outperforms DS-MML by relative improvements of 10.60%, 5.28% and 11.70% in terms of accuracy, AUC and recall, respectively.

Similarly, we report the performance comparison results of

Table 1: Performance comparison of our model and baseline methods on the ABIDE dataset using various evaluation metrics (%). Boldface numbers indicate the best classification performance.

Method	Accuracy	AUC	F1-score	Recall	Precision	κ	MCC
Logistic Regression	61.03	68.05	70.19	88.42	58.4	19.85	25.13
Gradient Boosting	59.97	62.04	62.24	63.48	61.34	19.53	19.67
GCN [14]	64.63	72.23	64.63	86.33	64.63	26.64	30.16
InceptionGCN [21]	72.69	72.81	79.27	-	-	-	-
EV-GCN [32]	80.83	84.98	81.24	-	-	-	-
LGL [22]	84.69	84.46	-	-	-	-	-
HI-GCN [26]	66.50	72.10	-	65.30	-	-	-
MMGL [15]	86.95	86.84	-	-	-	-	-
DeepGCN [16]	73.71	75.20	69.68	-	-	-	-
DS-MML [23]	87.62	92.00	-	86.76		-	-
AN-GCN (Ours)	**96.91**	**96.86**	**97.16**	**96.91**	**97.00**	**93.76**	**93.86**

our model and baseline methods on the ADNI dataset in Table 2, which also shows that AN-GCN performs better than all the competing baselines. Our model yields relative improvements of 15.61%, 8.66%, 14.13% and 15.70% over GCN in terms of accuracy, AUC, F1-score and precision, respectively. Moreover, AN-GCN significantly outperforms GCN in terms of recall, κ and MCC. In addition, our model outperforms the strongest baseline (i.e. BCN-GCN) by relative improvements of 5.76% and 4.36% in terms of accuracy and AUC, respectively.

In order to visually compare the performance of the proposed model to the baseline methods, we use box plots across all the folds on the ABIDE and ADNI dataset using accuracy and AUC as evaluation metrics, as shown in Figures 4 and 5. A box plot is a simple method for graphically depicting groups of numerical data through their quartiles, and it is commonly used to assess and compare the shape, central tendency, and variability of sample distributions, as well as to identify outliers. The box and whiskers show how the data is spread out. On each box, the central line represents the median, and the bottom and top edges of the box indicate the first and third quartile, respectively. The whiskers extend from the edges of the box to the lower and upper inner fences to show the range of the data. The fences are defined in terms of the inter-quartile range, and any value that falls outside the fences in considered as an outlier.

Figure 4 shows that our model outperforms the competing baselines in terms of the accuracy and AUC metrics for all the 10-folds. The higher the accuracy and AUC scores, the better the model distinguishes between patients suffering from ASD and healthy controls. As can be seen in Figure 4, the distribution of our model has less variability than GCN, gradient boosting and logistic regression in autism spectrum disorder prediction tasks. For instance, the median accuracy score for AN-GCN on the ABIDE dataset indicates significant difference in performance between our model and the three baseline methods. In addition, the box for AN-GCN is short, meaning that the accuracy values consistently hover around the average accuracy. However, the boxes for three baselines are taller, implying variable accuracy and AUC values compared to AN-GCN. We can also observe that the whisker is longer on the lower end of the box for GCN, indicating the distribution of both accuracy and AUC values is negatively skewed. For our AN-GCN model, the whisker lengths are short and roughly of the same length, indicating lower standard deviation and data symmetry, respectively.

Similarly, the box plots shown in Figures 5 indicate that our AN-GCN model outperforms the three baseline methods on the ADNI dataset in terms of both accuracy and AUC metrics. Interestingly, the box plot for GCN exhibits an outlier for accuracy values, as well as longer whiskers for AUC values. In addition, the box for the logistic regression is much taller than the other methods, indicating high variability in accuracy and AUC values. Gradient boosting also exhibits an outlier for AUC values.

We also evaluate the performance of our model against competing baselines using the precision-recall (PR) and receiver operating characteristic (ROC) curves on both ABIDE and ADNI datasets. The PR curve summarizes the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds. Precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned. A high area under the PR curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. Moreover, a PR curve that is closer to the upper left indicates a better performance. On the other hand, the ROC curve summarizes the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds. The area under the ROC curve (AUC) is a measure of discrimination in the sense that a model with a high AUC suggests that the model is able to accurately predict the value of an observation's response. Moreover, an ROC curve that is closer to the upper right indicates a better performance (i.e. true positive rate is higher than false positive rate).

Figures 6 and 7 show that our model yields the best performance compared to the baselines on both ABIDE and ADNI datasets. As can be seen, the PR (resp. ROC) curve of our model is much closer to the upper right (resp. left) than the cor-

Table 2: Performance comparison of our model and baseline methods on the ADNI dataset using various evaluation metrics (%). Boldface numbers indicate the best classification performance.

Method	Accuracy	AUC	F1-score	Recall	Precision	κ	MCC
Logistic Regression	58.71	51.61	68.80	58.71	59.67	04.48	04.04
Gradient Boosting	65.21	68.57	65.21	71.83	65.21	29.52	29.95
GCN [14]	84.98	89.32	84.89	58.82	84.98	60.37	62.58
HI-GCN [26]	75.40	75.60	-	66.40	-	-	-
BCN-GCN [33]	92.90	93.00	-	-	-	-	-
MMTGCN [24]	86.00	90.30	-	86.90	-	-	-
THS-GAN [34]	85.71	85.35	87.27	88.89	85.71	-	-
AN-GCN (Ours)	**98.25**	**97.06**	**96.89**	**98.25**	**98.33**	**95.68**	**95.84**

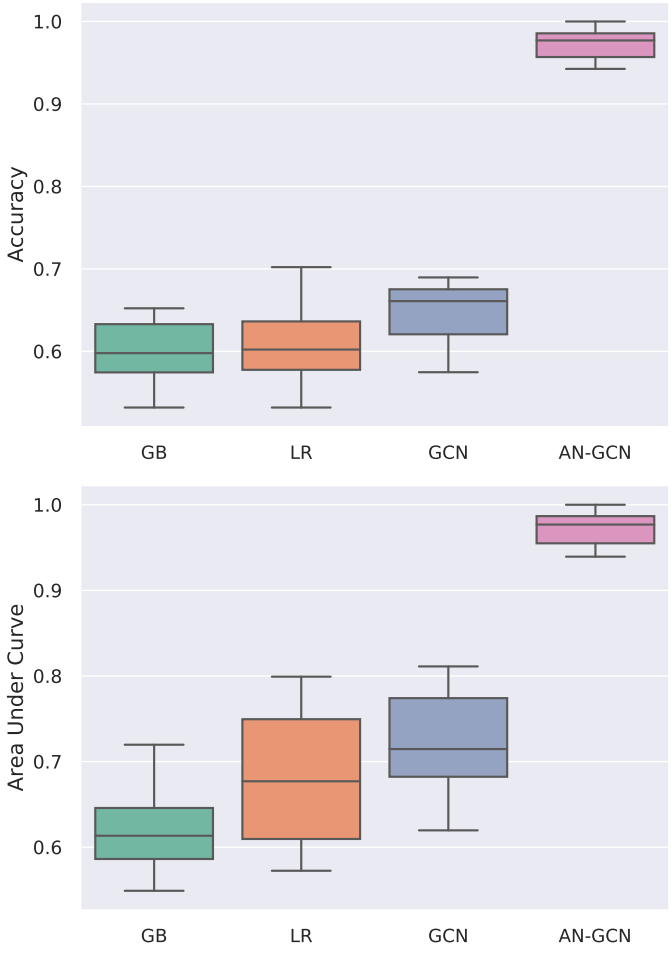

Figure 4: Comparative box plots of our model and baseline methods on the ABIDE dataset using accuracy and AUC scores over all cross-validation folds.

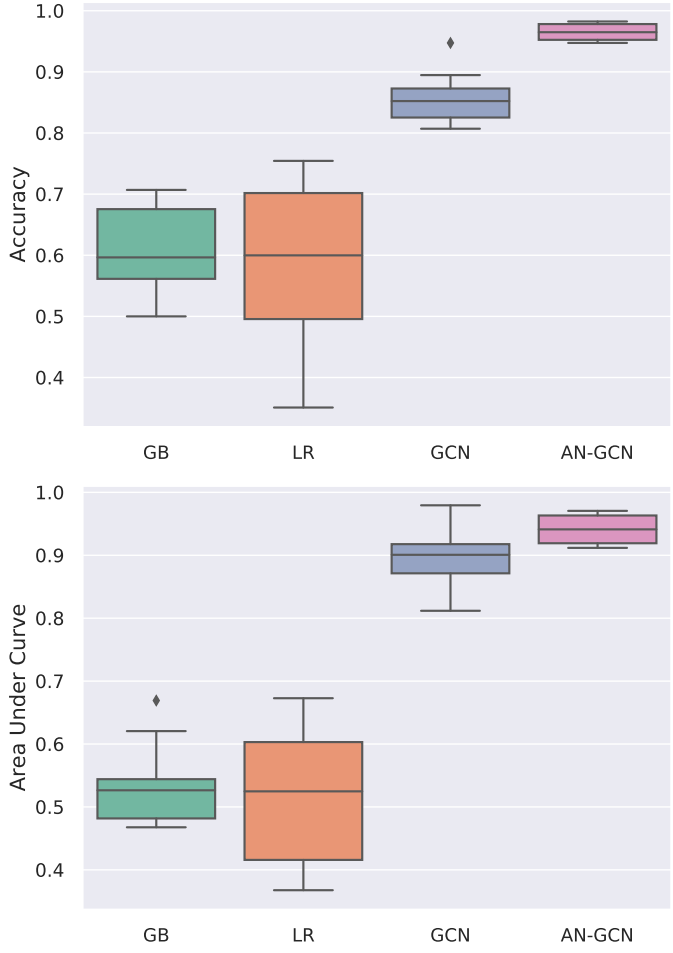

Figure 5: Comparative box plots of our model and baseline methods on the ADNI dataset using accuracy and AUC scores over all cross-validation folds.

responding curves for the baselines, indicating the better performance of AN-GCN in disease prediction tasks. In the ROC curves, the diagonal dashed line, which depicts a random algorithm (i.e., random guessing of classes), divides the ROC space. Points above the diagonal represent good classification results (better than random), points below the line poor results (worse than random). Notice that the ROC curves of the logistic regression and gradient boosting are closer to the diagonal line on the ABIDE dataset, indicating poor classification performance.

Overall, our AN-GCN model outperforms GCN and the

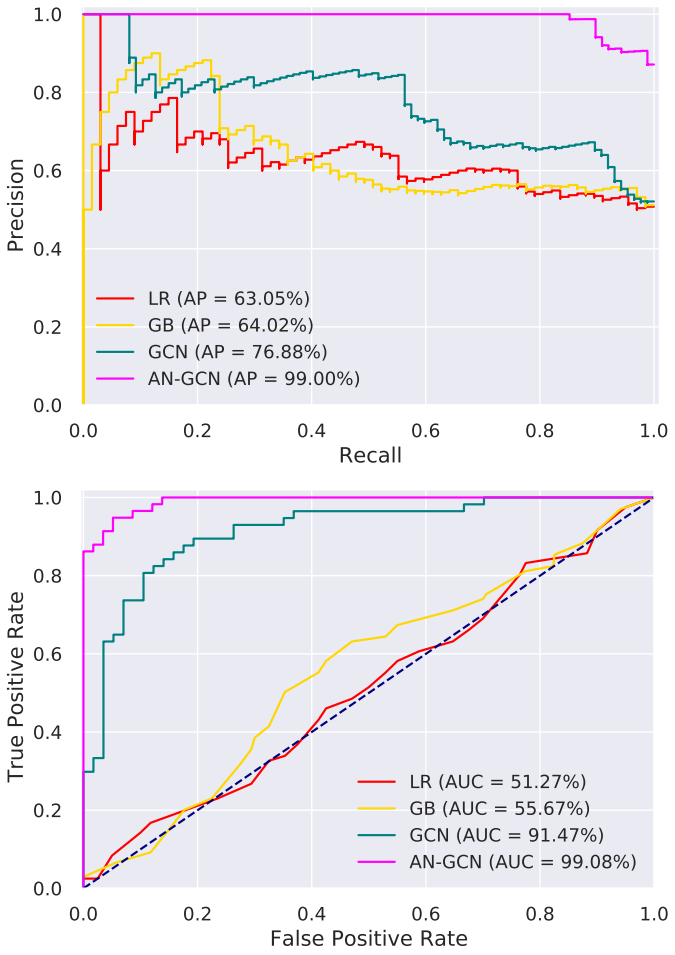

Figure 6: Precision-Recall and ROC curves of our model and baseline methods on the ABIDE dataset. Average precision (AP) and AUC values are enclosed in parentheses.

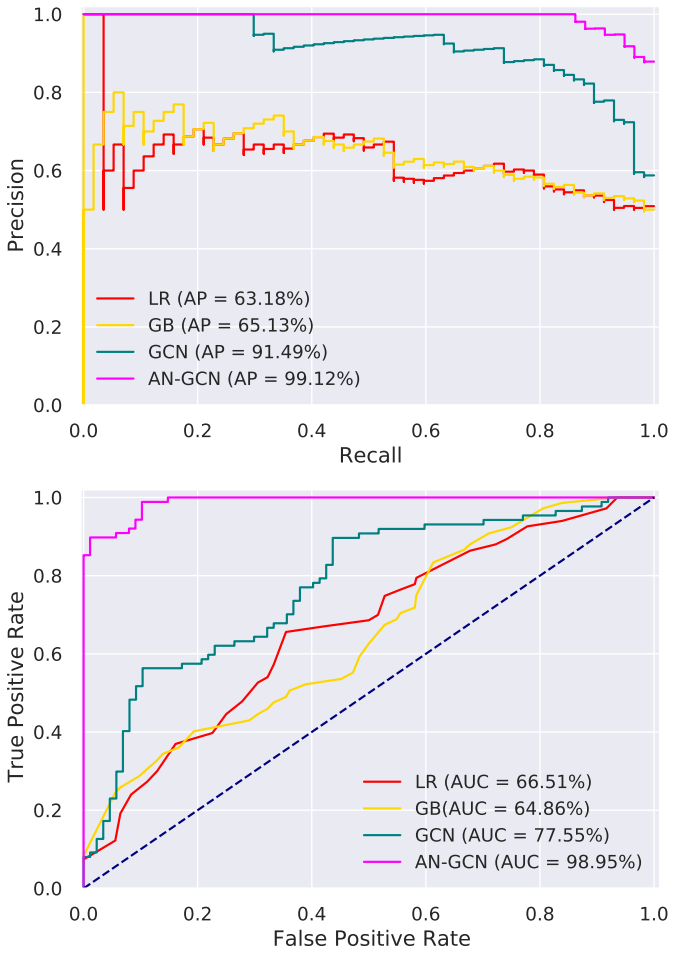

Figure 7: Precision-Recall and ROC curves of our model and baseline methods on the ADNI dataset. Average precision (AP) and AUC values are enclosed in parentheses.

other baselines significantly and consistently across all datasets, achieving state-of-the-art results in terms of various performance evaluation metrics. In particular, our model improves over the GCN baseline by a big margin. Another key observation is that AN-GCN also outperforms DeepGCN, yielding relative improvements of 31.47%, 28.80% and 39.44% over GCN in terms of accuracy, AUC and F1-score, respectively, on the ABIDE dataset.

4.3 Parameter Sensitivity Analysis

In order to answer **RQ2** and **RQ3**, we analyze the sensitivity of our disease prediction model to the choice of the number of network layers and the batch size. As the number of network layers plays an important role, we first study how the performance changes as a function of the network depth. Then, we study the performance variation for our model with respect to the batch size on both ABIDE and ADNI datasets.

Mitigating the Oversmoothing Problem. To evaluate the robustness of our approach to oversmoothing, we study the performance variation for our multi-layer AN-GCN model on the ABIDE and ADNI datasets with respect to the number of layers. Figure 8 shows how the node classification accuracy changes with the network's depth. As can be seen, the performance of AN-GCN does not significantly degrade compared to GCN when the number of layers increases. Moreover, the performance gap between AN-GCN and GCN becomes quite noticeable when the network's depth rises. Hence, the classification performance of AN-GCN remains relatively stable as we increase the number of layers, demonstrating the robustness of our model against oversmoothing. This is largely due to the fact that the aggregation scheme of the proposed approach leverages residual connections to help alleviate the oversmoothing problem.

Effect of Batch Size. We test the performance of our model using different values for the batch size on the ABIDE and ADNI datasets. As shown in Figure 9, the classification accuracy increases rapidly at the beginning (i.e. for smaller batch sizes), reaching the highest value when the batch size is equal to 1000, and then slowly starts to decline on the ABIDE dataset or slows down on the ADNI dataset. This indicates that the batch size also plays an important role. In fact, we can observe that

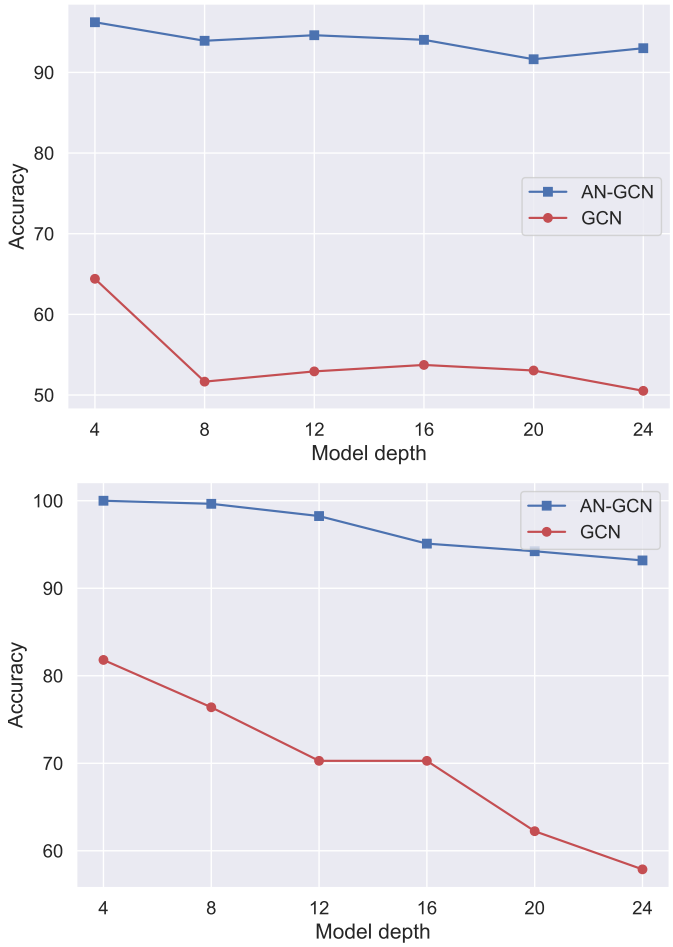

Figure 8: Performance comparison of AN-GCN and GCN on the ABIDE (top) and ADNI (bottom) datasets as we increase the number of layers.

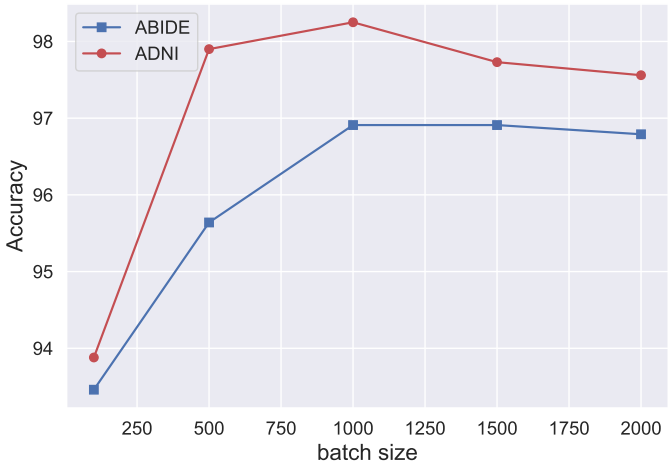

Figure 9: Sensitivity analysis of our model to the batch size on the ABIDE and ADNI datasets.

using a large batch size to train our model allows computational speedups from the parallelism of GPUs, but a larger batch size leads to poor generalization. It should also be pointed out that the drawback of using a smaller batch size is that the model is not guaranteed to converge to the global optimum.

5 Conclusion

In this paper, we introduced an end-to-end graph convolutional aggregation model by learning discriminative node representations from a population graph, consisting of subjects as nodes and edges as connections between subjects, with the goal to predict the status of each subject (i.e. diseased or healthy control) using imaging and non-imaging features associated to the graph nodes and edges, respectively. The proposed framework leverages skip connections and identity mapping, as well as aggregation in graph sampling in a bid to alleviate the problem of over-smoothing in graph convolutional networks. We demonstrated through extensive experiments that our model outperforms existing graph convolutional based methods for disease prediction on two large benchmark datasets, achieving significant relative improvements in classification accuracy over GCN and other strong baselines. For future work, we plan to integrate higher-order graph convolutions into our model with the aim to capture long-range dependencies between subjects in a population graph. We would also like to investigate the tradeoff introduced by the hyperparameters of the layer-wise propagation rule of our model with the purpose of gaining more theoretical insight. In addition, we intend to apply our model to data relational graphs, where nodes can be connected to each other via multiple relations.

References

[1] T. R. Insel and B. N. Cuthbert, "Brain disorders? precisely," *Science*, pp. 499–500, 2015.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[3] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. International Conference on Machine Learning*, pp. 6861–6871, 2019.

[4] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "GraphSAINT: Graph sampling based inductive learning method," in *International Conference on Learning Representations*, 2020.

[5] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. International Conference on Machine Learning*, pp. 1725–1735, 2020.

[6] M. Khosla, K. Jamison, G. H. Ngo, A. Kuceyeski, and M. R. Sabuncu, "Machine learning in resting-state fmri analysis," *Magnetic Resonance Imaging*, pp. 101–121, 2019.

[7] K. Gopinath, C. Desrosiers, and H. Lombaert, "Graph convolutions on spectral embeddings for cortical surface parcellation," *Medical Image Analysis*, pp. 297–305, 2019.

[8] C. Su, J. Tong, Y. Zhu, P. Cui, and F. Wang, "Network embedding in biomedical data science," *Briefings in Bioinformatics*, pp. 182–197, 2020.

[9] X. Yue, Z. Wang, J. Huang, S. Parthasarathy, S. Moosavinasab, Y. Huang, S. M. Lin, W. Zhang, P. Zhang, and H. Sun, "Graph embedding on biomedical networks: methods, applications and evaluations," *Bioinformatics*, pp. 1241–1251, 2020.

[10] J. Yang, Q. Zhu, R. Zhang, J. Huang, and D. Zhang, "Unified brain network with functional and structural data," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 114–123, 2020.

[11] L. Goldsberry, W. Huang, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro, "Brain signal analytics from graph signal processing perspective," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 851–855, 2017.

[12] S. I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, and D. Rueckert, "Metric learning with spectral graph convolutions on brain connectivity networks," *NeuroImage*, pp. 431–442, 2018.

[13] G. Ma, N. K. Ahmed, T. L. Willke, D. Sengupta, M. W. Cole, N. B. Turk-Browne, and P. S. Yu, "Deep graph similarity learning for brain data analysis," in *Proc. ACM International Conference on Information and Knowledge Management*, pp. 2743–2751, 2019.

[14] S. Parisot, S. I. Ktena, E. Ferrante, M. Lee, R. Guerrero, B. Glocker, and D. Rueckert, "Disease prediction using graph convolutional networks: application to autism spectrum disorder and Alzheimer's disease," *Medical Image Analysis*, pp. 117–130, 2018.

[15] S. Zheng, Z. Zhu, Z. Liu, Z. Guo, Y. Liu, and Y. Zhao, "Multi-modal graph learning for disease prediction," *arXiv preprint arXiv:2107.00206*, 2021.

[16] M. Cao, M. Yang, C. Qin, X. Zhu, Y. Chen, J. Wang, and T. Liu, "Using deepGCN to identify the autism spectrum disorder from multi-site resting-state data," *Biomedical Signal Processing and Control*, p. 103015, 2021.

[17] B. Xu, H. Shen, Q. Cao, Y. Qiu, and X. Cheng, "Graph wavelet neural network," in *International Conference on Learning Representations*, 2019.

[18] Q. Li, Z. Han, and X. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *AAAI Conference on Artificial Intelligence*, pp. 3538–3545, 2018.

[19] K. Xu, C. Li, Y. Tian, T. Sonobe, K. ichi Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. International Conference on Machine Learning*, 2018.

[20] L. Zhao and L. Akoglu, "PairNorm: Tackling oversmoothing in GNNs," in *International Conference on Learning Representations*, 2020.

[21] A. Kazi, S. shekarforoush, S. krishna, H. Burwinkel, G. Vivar, K. Kortuem, S.-A. Ahmadi, S. Albarqouni, and N. Navab, "InceptionGCN: Receptive field aware graph convolutional network for disease prediction," in *Proc. International Conference on Information Processing in Medical Imaging*, 2019.

[22] L. Cosmo, A. Kazi, S.-A. Ahmadi, N. Navab, and M. Bronstein, "Latent-graph learning for disease prediction," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 643–653, 2020.

[23] L. Pan, J. Liu, M. Shi, C. W. Wong, and K. H. K. Chan, "Identifying autism spectrum disorder based on individual-aware down-sampling and multi-modal learning," *arXiv preprint arXiv:2109.09129*, 2021.

[24] D. Yao, J. Sui, M. Wang, E. Yang, Y. Jiaerken, N. Luo, P. T. Yap, M. Liu, and D. Shen, "A mutual multi-scale triplet graph convolutional network for classification of brain disorders using functional or structural connectivity," *IEEE Transactions on Medical Imaging*, pp. 1279–1289, 2021.

[25] Y. Rong, W. Huang, T. Xu, and J. Huang, "DropEdge: Towards deep graph convolutional networks on node classification," in *International Conference on Learning Representations*, 2020.

[26] H. Jiang, P. Cao, M. Xu, J. Yang, and O. Zaiane, "HI-GCN: A hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction," *Computers in Biology and Medicine*, vol. 127, pp. 1–16, 2020.

[27] Y. Huang and A. C. S. Chung, "Diffusion improves graph learning," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 13354–13366, 2019.

[28] Y. Chu, X. Wang, Q. Dai, Y. Wang, Q. Wang, S. Peng, X. Wei, J. Qiu, D. R. Salahub, Y. Xiong, *et al.*, "MDA-GCNFTG: identifying miRNA-disease associations based on graph convolutional networks via graph sampling through the feature and topology graph," *Briefings in Bioinformatics*, pp. 1–19, 2021.

[29] J. C. Paetzold, J. McGinnis, S. Shit, I. Ezhov, P. Büschl, C. Prabhakar, M. I. Todorov, A. Sekuboyina, G. Kaissis, A. Ertürk, *et al.*, "Whole brain vessel graphs: A dataset and benchmark for graph learning and neuroscience (vesselgraph)," *arXiv preprint arXiv:2108.13233*, 2021.

[30] Y. Li and Y. Yuan, "Convergence analysis of two-layer neural networks with relu activation," in *Advances in Neural Information Processing Systems*, pp. 597–607, 2017.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[32] Y. Huang and A. C. S. Chung, "Edge-variational graph convolutional networks for uncertaintyaware disease prediction," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 562–572, 2020.

[33] P. Gu, X. Xu, Y. Luo, P. Wang, and J. Lu, "BCN-GCN: A novel brain connectivity network classification method via graph convolution neural network for Alzheimer's disease," in *Proc. International Conference on Neural Information Processing*, pp. 657–668, 2021.

[34] W. Yu, B. Lei, M. K. Ng, A. C. Cheung, Y. Shen, and S. Wang, "Tensorizing GAN with high-order pooling for Alzheimer's disease assessment," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.