FSNet: Dual Interpretable Graph Convolutional Network for Alzheimer's Disease Analysis

Hengxin Li, Xiaoshuang Shi, Xiaofeng Zhu, *Senior Member, IEEE*, Shuihua Wang, *Senior Member, IEEE*, and Zheng Zhang, *Senior Member, IEEE*

***Abstract*— Graph Convolutional Networks (GCNs) are widely used in medical images diagnostic research, because they can automatically learn powerful and robust feature representations. However, their performance might be significantly deteriorated by trivial or corrupted medical features and samples. Moreover, existing methods cannot simultaneously interpret the significant features and samples. To overcome these limitations, in this paper, we propose a novel dual interpretable graph convolutional network, namely FSNet, to simultaneously select significant features and samples, so as to boost model performance for medical diagnosis and interpretation. Specifically, the proposed network consists of three modules, two of which leverage one simple yet effective sparse mechanism to obtain feature and sample weight matrices for interpreting features and samples, respectively, and the third one is utilized for medical diagnosis. Extensive experiments on the Alzheimer's Disease Neuroimaging Initiative (ADNI) datasets demonstrate the superior classification performance and interpretability over the recent state-of-the-art methods.**

***Index Terms*—Alzheimer's disease diagnosis research, feature interpretability, graph convolutional network, sample interpretability.**

I. Introduction

ALZHEIMER'S disease (AD) is one of the most common clinical neurodegenerative diseases, which can severely degrade the quality of life of the elderly [7], [10]. Unfortunately, now there is no fundamental medical treatment to cure AD. Since early intervention is an effective way to slow down its deterioration, neuroimaging techniques, like Magnetic Resonance Imaging (MRI), are widely used for early diagnosis by providing accurate information about the state of the brain [1], [19]. However, manually examining neuroimaging data is laborious and time consuming. Thus, machine learning techniques, which are promising to reduce the workload of neurologists in the future [6], have been widely employed for medical images diagnostic research.

Since each feature might have a different contribution for data classification [28], [35], traditional machine learning methods, like Random Forest [27], rank the importance of features for classification or select significant features to interpret classification results. But they usually provide poor diagnostic performance due to the lack of ability on extracting powerful feature representations, thereby leading to unconvincing interpretation results [33]. Deep learning methods have been demonstrated the powerful capability on extracting discriminative feature representations with a large amount of high quality training data [5], [39]. However, most deep learning methods cannot directly interpret the significance of features, and thus fail to provide interpretable diagnosis results, which is one major concern in the medical domain. Additionally, it is difficult to obtain a large number of data with accurate labels and clean features in medical domain, due to expensive costs and environmental diversity of data acquisition and subjective assessment of clinicians [18], which would decrease the performance of deep learning methods. Therefore, it is a very challenging task to employ a small number of medical training samples to attain interpretable and robust results for deep learning methods.

Graph Convolutional Networks (GCN) is a widely used deep learning framework for Alzheimer's disease diagnosis research and analysis, because they simultaneously consider semantic information and structure information, and can produce more accurate classification results than traditional machine learning methods on a small number of training samples [19], [20], [31]. However, GCNs cannot directly interpret the significance of features, and thus it usually utilizes the post-hoc interpretable strategy, which discovers significant features using an explanatory model after the well-trained classification model, thereby consuming more time and possibly causing inferior classification and interpretability performance [36]. To overcome these limitations, most recently, [15] proposed an interpretable framework to discover the most significant features during model training. But this framework neglects features weights, which are also very significant for data classification.

Another limitation of most existing interpretable GCNs is that they do not consider the different significance of training

Manuscript received December 30, 2021; revised May 7, 2022; accepted June 3, 2022. *(Hengxin Li and Xiaoshuang Shi had equivalent contribution to this work.) (Corresponding authors: Xiaofeng Zhu, Shuihua Wang, and Zheng Zhang.)*

Hengxin Li, Xiaoshuang Shi, and Xiaofeng Zhu are with the Center for Future Media and School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: lihengxin7@gmail.com; xsshi2021@uestc.edu.cn; seanzhuxf@gmail.com).

Shuihua Wang is with the School of Computing and Mathematics Sciences, University of Leicester, LE1 7RH Leicester, U.K. (e-mail: shuihuawang@ieee.org).

Zheng Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 150001, China (e-mail: darrenzz219@gmail.com).

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by ADNI.

Digital Object Identifier 10.1109/TETCI.2022.3183679

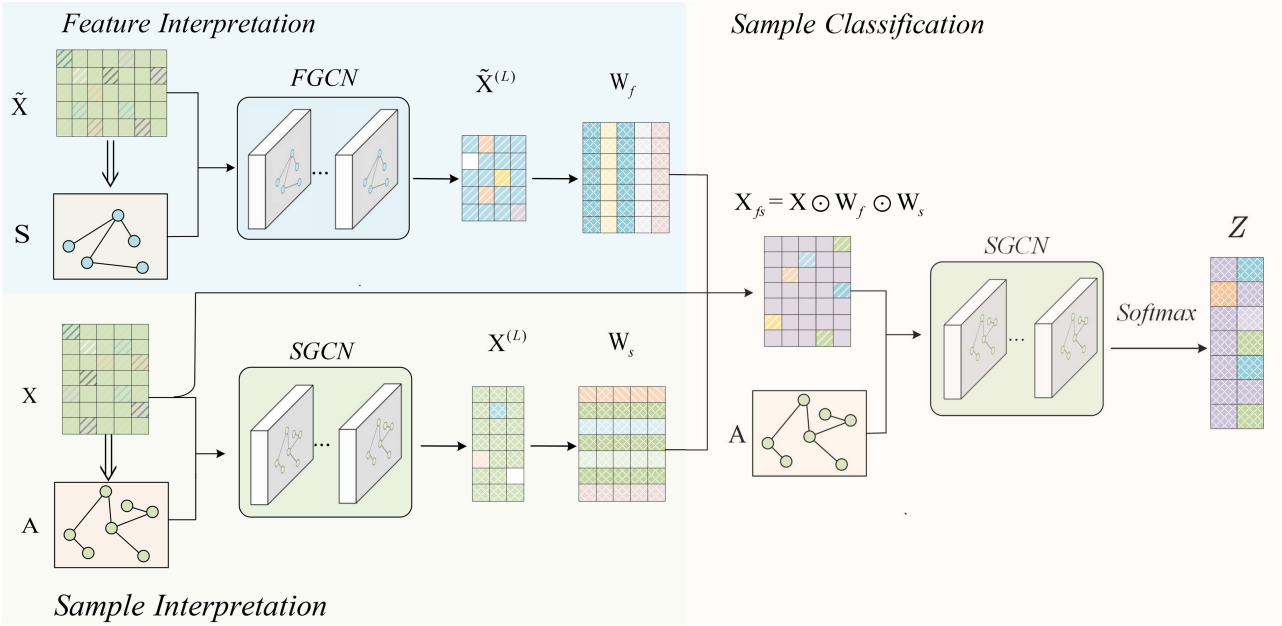


Fig. 1. The architecture of our proposed FSNet with three modules: feature interpretation, sample interpretation and classification. The feature and sample interpretation modules aim to attain sparse weight matrices W_f and W_s to interpret the significance of features and samples, respectively. The sample classification module only employs the significant features and samples for classification. Note that the sample interpretation and classification modules share the same parameters. All these three modules are jointly conducted to obtain AD diagnostic results in an end-to-end learning manner.

samples, which usually contain corrupted labels or features, thereby deteriorating model performance. Although several deep learning methods employ self-paced learning to guide model training by ranking the significance of training samples [14], [24], [32], they usually require some clean samples to assist in training [21] to attain good performance. This might restrict their applications. Additionally, they fail to simultaneously take the feature significance into consideration, leading to the sub-optimal model performance. Therefore, it is very necessary to simultaneously interpret the significance of features and samples during model training.

In this paper, we propose a novel dual interpretable graph convolutional network, namely FSNet, to address the aforementioned issues, *i.e.*, simultaneously interpreting the significance of features and samples during model training. For clarity, we present the architecture of FSNet in Fig. 1. Specifically, it utilizes two sub-networks corresponding to two modules, FGCN and SGCN, to obtain sparse feature and sample weight matrices for selecting significant features and samples, respectively. Then, the third module utilizes SGCN for classification.

We summarize three major contributions of this paper as follows:

- We propose a novel graph convolutional network to simultaneously interpret the significance of both features and samples in early-stage AD diagnosis.
- Different from previous interpretation methods without considering the weights of features or samples, we propose a simple yet effective interpretation mechanism with selecting significant features or samples and meanwhile assigning weights to them. Additionally, experiments demonstrate that considering the weights is beneficial to AD diagnosis.
- Extensive experiments on four AD datasets demonstrate that the proposed network outperforms recent state-of-the-art methods on classification and interpretation performance.

The rest of the paper is organized as follows. Section II introduces the proposed dual interpretable graph convolutional network; Section III shows and analyzes experimental results of various methods; Finally, Section IV concludes this paper and points out the future work.

II. Method

In this section, we will introduce the proposed dual interpretable graph convolutional network, FSNet. As shown in Fig. 1, FSNet has three modules, specifically, feature interpretation adopts FGCN to attain a sparse feature weight matrix, sample interpretation utilizes SGCN to attain a sparse sample weight matrix, and sample classification employs SGCN for sample classification. Additionally, in this paper, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n\times d}$ represents a matrix with n samples, $\mathbf{x}_i \in \mathbb{R}^d$ $(1 \le i \le n)$ denotes the i-th sample with d features, and x_{ik} is the k-th feature of the i-th sample.

A. Feature Interpretation

To generate feature weight matrix $\mathbf{W}_f \in \mathbb{R}^{n\times d}$ for interpreting the significance of features, differing from that using the sample as nodes, we innovatively consider each feature as one node and obtain $\mathbf{W}_f$ by the following steps.

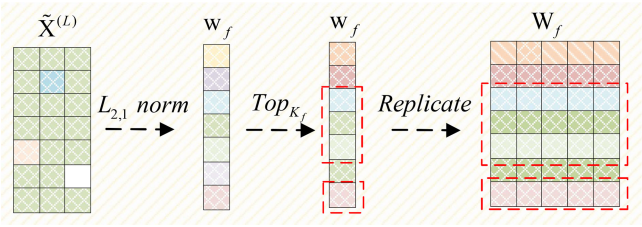


Fig. 2. Procedure for obtaining the sparse feature weight matrix and the red dashed boxes indicate the selected important features. Specifically, $\tilde{\mathbf{X}}_n^{(L)} \in \mathbb{R}^{d\times n_L}$ is the embedding matrix with d features after graph convolution, and then for each row vector, we utilize the $L_{2,1}$-norm to obtain the weight vector $w_f \in \mathbb{R}^d$. Finally, w_f is considered as a row vector and then replicated n times by rows to attain the final feature weight matrix $\mathbf{W}_f \in \mathbb{R}^{n\times d}$.

Specifically, we transpose $\mathbf{X}$ to obtain $\tilde{\mathbf{X}}$, *i.e.*, $\tilde{\mathbf{X}} = \mathbf{X}^T \in \mathbb{R}^{d\times n}$, and then utilize each feature as the node to generate an adjacency matrix $\mathbf{S} \in \mathbb{R}^{d\times d}$ to describe the correlations among features. The specific generation steps from a feature matrix $\tilde{\mathbf{X}}$ to the adjacency matrix $\mathbf{S}$ can be found in (10) shown in Section III. After obtaining $\mathbf{S}$, given an L-layer graph convolutional network, FGCN, which is specifically used to aggregate the neighbor information of each feature. Let $\tilde{\mathbf{X}}^{(0)} = \tilde{\mathbf{X}}$ denote the input of the network and feed it with $\mathbf{S}$ into the network to obtain a feature embedding representation $\tilde{\mathbf{X}}^{l+1}$, which can be represented by:

$$\tilde{\mathbf{X}}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-1/2}\hat{\mathbf{S}}\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{X}}^{(l)}\mathbf{\Theta}_f^{(l)}), \tag{1}$$

where $l = 0, \ldots, L-1$, $\tilde{\mathbf{X}}^{l+1}$ and $\tilde{\mathbf{X}}^{(l)} \in \mathbb{R}^{d\times n_{l+1}}$ are the output of the $l+1$-th and l-th layer, respectively, $\hat{\mathbf{S}} = \mathbf{S} + \mathbf{I}_d$, $\tilde{\mathbf{D}} = diag(\tilde{\mathbf{d}}_1, \ldots, \tilde{\mathbf{d}}_d)$ is a diagonal matrix with $\tilde{\mathbf{d}}_i = \sum_{j=1}^{n}(\hat{\mathbf{S}}_{ij})$, $\mathbf{\Theta}_s^{(l)} \in \mathbb{R}^{n_l\times n_{l+1}}$ represents the parameters of the $l+1$-th layer, and $\sigma(\cdot)$ represents an activation function.

After the L-layer graph convoluition, we can obtain an embedding representation $\tilde{\mathbf{X}}^{(L)} = [\tilde{\mathbf{x}_0}^{(L)}, \tilde{\mathbf{x}}_1^{(L)}, \ldots, \tilde{\mathbf{x}}_n^{(L)}] \in \mathbb{R}^{d\times n_L}$ and then we can utilize $\tilde{\mathbf{X}}^{(L)}$ to generate $\mathbf{W}_f$.

Here, our goal is to obtain the weight of each feature based on its embedding representation $\tilde{\mathbf{X}}^{(L)}$, and when the i-th feature is trivial, its corresponding weight should be very small or even zero. Under such conditions, the $L_{2,0}$-norm is the most suitable constraint for removing the redundant features by controlling sparsity while maintaining as much original information as possible [9]. However, the $L_{2,0}$-norm is non-convex and non-differential so that it is very difficult to optimize [4]. Hence, following the same strategy in [34], we utilize the $L_{2,1}$-norm to calculate the weight of each feature, because the L_{21}-norm is the minimum convex hull of the $L_{2,0}$-norm and it can also encourage row sparsity.

The procedure has been shown in Fig. 2. Firstly, let $\mathbf{w}_f = [w_{f,1}, w_{f,2}, \ldots, w_{f,d}] \in \mathbb{R}^d$ be a weight vector, where $w_{f,i} \in \mathbf{w}_f$ is the weight of the i-th feature, we calculate $w_{f,i}$ by employing the L_{21}-norm as follows:

$$w_{f,i} = \sqrt{\sum_{j=1}^{n_L}(\tilde{x}_{ij}^{(L)})^2}. \tag{2}$$

Secondly, in order to find the significant features and then only adopt them to optimize the model for training, we select K_f features corresponding to the top K_f weights and set the weights of the remaining features as 0. Hence, the weight for the i-th node is updated as:

$$w_{f,i} = \begin{cases} w_{f,i} \text{ if } w_{f,i} > w_{f,K_f} \\ 0 \quad \text{otherwise,} \end{cases} \tag{3}$$

where w_{f,K_f} is the K_f-th largest weight in $\mathbf{w}_f$.

As we can see, (3) can maintain the significant features and their weights and meanwhile remove the trivial features (redundancy or noise) during model training.

Thirdly, we utilize the weight vector $\mathbf{w}_f$ as a row vector and replicate it n times to obtain a feature weight matrix $\mathbf{W}_f \in \mathbb{R}^{n\times d}$ with the same size as the matrix $\mathbf{X}$ for subsequent model training. For example,

$$\mathbf{W}_f = \begin{bmatrix} w_{f,1} & w_{f,2} & \ldots & w_{f,d} \\ w_{f,1} & w_{f,2} & \ldots & w_{f,d} \\ \ldots & \ldots & \ddots & \ldots \\ w_{f,1} & w_{f,2} & \ldots & w_{f,d} \end{bmatrix} \tag{4}$$

For clarity, we present the procedure to obtain the sparse weight matrix $\mathbf{W}_f$ in Fig. 2.

B. Sample Interpretation

Because training samples also have different significance, here, we present the process for sample interpretation. Given an L-layer graph convolutional network, SGCN, the matrix $\mathbf{X}^{(0)} = \mathbf{X} \in \mathbb{R}^{n\times d}$ and the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ containing the sample correlations, we conduct graph convolution to aggregate feature representations among their neighbors. The process can be formulated as:

$$\mathbf{X}^{(l+1)} = \sigma(\mathbf{D}^{-1/2}\hat{\mathbf{A}}\mathbf{D}^{-1/2}\mathbf{X}^{(l)}\mathbf{\Theta}_s^{(l)}), \tag{5}$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$, $\mathbf{D} = diag(\mathbf{d}_1, \ldots, \mathbf{d}_n)$ is a diagonal matrix with $\mathbf{d}_i = \sum_{j=1}^{n}(\hat{\mathbf{A}}_{ij})$, and $\mathbf{\Theta}_s^{(l)} \in \mathbb{R}^{d_l\times d_{l+1}}$ represents the parameters of the $l+1$-th layer.

Next, we can obtain the sample weight matrix $\mathbf{W}_s \in \mathbb{R}^{n\times d}$ by using the same three steps as that attaining $\mathbf{W}_f$. Specifically, we first obtain a sample weight vector $\mathbf{w}_s = [w_{s,1}, w_{s,2}, \ldots, w_{s,n}] \in \mathbb{R}^n$ based on (2); Then, same as (3), we can update $\mathbf{w}_s$ to retain K_s samples with top K_s weights; Finally, we replicate the updated weight vector $\mathbf{w}_s$ by d times and then transpose it to obtain the sample weight matrix $\mathbf{W}_s \in \mathbb{R}^{n\times d}$.

In summary, our proposed method simultaneously considers the feature information of each sample and the structural information among samples to obtain the sample weight matrix. Meanwhile, it can filter out the trivial samples (*e.g.*, samples with corrupted labels or features).

C. GCN Classification

There exists many node classification methods, such as GCN [12], GraphSage [8], GAT [30], but they treat each node and their features equally, thereby possibly leading to sub-optimal results in many cases. Differing from them, the proposed

network simultaneously takes into account the weights of both features and samples during model training, in order to obtain superior performance.

In this paper, we perform Hadamard product operation on the feature weight matrix $\mathbf{W}_f$ and the sample weight matrix $\mathbf{W}_s$ with the input $\mathbf{X}$ to obtain a new matrix $\mathbf{X}_{fs} \in \mathbb{R}^{n\times d}$, which is calculated by:

$$\mathbf{X}_{fs} = \mathbf{X} \odot \mathbf{W}_f \odot \mathbf{W}_s, \tag{6}$$

where $\odot$ denotes the Hadamard product operation, *i.e.*, two matrices are multiplied element by element. Each feature in one sample has its own weight, and then we feed $\mathbf{X}_{fs}$ and the adjacency matrix $\mathbf{A}$ into a graph convolutional network.

Because using two completely different networks to extract feature representations will increase the model complexity with consuming more training costs. To reduce the model complexity and training time costs, we utilize SGCN for classification. This means $\mathbf{X}_{fs}$ and $\mathbf{X}$ are fed into the same sub-network. Hence, in the $l+1$-th layer, its output matrix $\mathbf{X}_{fs}^{(l+1)}$ can be formulated as:

$$\mathbf{X}_{fs}^{(l+1)} = \sigma(\mathbf{D}^{-1/2}\hat{\mathbf{A}}\mathbf{D}^{-1/2}\mathbf{X}_{fs}^{(l)}\Theta_s^{(l)}), \tag{7}$$

where the parameters have the same meaning as (5). $\mathbf{X}_{fs}$ can share the graph convolution with $\mathbf{X}$, because it is just the result of adding weights onto $\mathbf{X}$ (see (7)).

After L graph convolutional layers, we can obtain the output matrix $\mathbf{X}_{fs}^L = [\mathbf{x}_0^L, \mathbf{x}_1^L, \ldots, \mathbf{x}_n^L] \in \mathbb{R}^{n\times c}$, and then obtain their predicted label probabilities by using the softmax function, e.g., $\mathbf{Z} = [\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{n-1}] \in \mathbb{R}^{n\times c}$, where $\mathbf{z}_i \in \mathbb{R}^c$ denotes the label prediction of the i-th sample and z_{ij} is its probability belonging to the j-th class. It is calculated as follows:

$$z_{ij} = \mathrm{softmax}_j\left(\mathbf{x}_{ij}^L\right) = \frac{\exp\left(\mathbf{x}_{ij}^L\right)}{\sum_{m\in n}\exp\left(\mathbf{x}_{im}^L\right)}. \tag{8}$$

Finally, we employ the cross-entropy function for model training, it is:

$$Loss = -\sum_{i\in N}\sum_{j=1}^{c} y_{ij}\ln z_{ij}, \tag{9}$$

where $\mathcal{N}$ is defined as the set of labeled samples. By minimizing (10), we can obtain the optimal graph convolutional parameters $\mathbf{\Theta}_f = \{\mathbf{\Theta}_f^0, \ldots, \mathbf{\Theta}_f^L\}$ and $\mathbf{\Theta}_s = \{\mathbf{\Theta}_s^0, \ldots, \mathbf{\Theta}_s^L\}$. For clarity, we present the detailed procedure of the proposed FSNet in Algorithm 1.

Algorithm 1: FSNet.

Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\tilde{\mathbf{X}} \in \mathbb{R}^{d\times n}$, label information $\mathbf{Y}$, and hyperparameters K_f, K_s and T.

Initialization: GCN parameters $\mathbf{\Theta}_f$, $\mathbf{\Theta}_s$.
while $epoch < T$ **do**
  $\tilde{\mathbf{X}}^{l+1} \leftarrow \{\tilde{\mathbf{X}}, \mathbf{S}, \mathbf{\Theta}_f\}$ by (1);
  $\mathbf{W}_f \leftarrow \{\tilde{\mathbf{X}}^{l+1}, K_f\}$ by (2), (3) and (4);
  $\mathbf{X}^{l+1} \leftarrow \{\mathbf{X}, \mathbf{A}, \mathbf{\Theta}_s\}$ by (5);
  $\mathbf{W}_s \leftarrow \{\mathbf{X}^{l+1}, K_s\}$ by (2), (3) and (4);
  $\mathbf{X}_{fs} \leftarrow \{\mathbf{X}, \mathbf{W}_f, \mathbf{W}_s\}$ by (6);
  $\mathbf{Z} \leftarrow \{\mathbf{X}_{fs}, \mathbf{A}, \mathbf{\Theta}_s\}$ by (7);
  $Loss \leftarrow \{\mathbf{Y}, \mathbf{Z}\}$ by (8);
  Back-propagate $Loss$ to update model parameters;
end while

Output: $\mathbf{Z}$, $\mathbf{W}_f$, $\mathbf{W}_s$.

III. Experiments

A. Experimental Setup

1) Datasets Description: Raw digital images were downloaded from the ADNI database. All MRI images used in our experiments were 1.5 T T1-weighted MRI data. Firstly, those raw images pre-processed by the way described in [23], including removing non-brain tissue, correcting for motion and time, registration, filtering and smoothing. After obtaining the pre-processed MRI images, we further segmented those images into three different tissues: gray matter, white matter, and cerebrospinal fluid, and then warped them into Jacob template [11] to obtain 93 brain regions. Finally, we extracted the gray matter volume of each brain region as one feature and in this way each subject (patient) could be represented by a 93-dimensional feature vector.

TABLE I
The Number of Sample in ADNI Datasets

Datasets	Samples	Datasets	Samples
AD-NC	186:226	AD-MCI	186:393
NC-MCI	226:393	MCIn-MCIp	226:167

In our experiments, we totally processed 805 subjects, which included 186 AD patients, 393 Mild Cognitive Impairment (MCI) patients, and 226 normal controls (NC). Furthermore, 393 MCI patients included 226 MCI converters (MCIp) and 167 MCI non-converters (MCIn). Next, we divided these samples into four binary datasets (*i.e.*, AD-NC, AD -MCI, NC-MCI, and MCIn-MCIp). The number of samples of four binary datasets is summarized in Table I.

2) Graph Construction: Considering that the patients with the same class might contain some similar characteristics, the diagnostic study of Alzheimer's disease in [40] utilizes both the feature information and the structural information to produce better classification performance. Structural information refers to the connections between patients and is usually represented as an adjacency matrix. However, in this paper, the initial input $\mathbf{X}$ only contains the features of each patient without considering their correlations. Here, we not only construct the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ to describe the relations among samples, but also build the adjacency matrix $\mathbf{S} \in \mathbb{R}^{d\times d}$ to describe the relations among features.

To calculate the similarity between any two different features, we utilize the inverse of the distance between the i-th and j-th

features as follows:

$$s_{ij} = \frac{1}{\sqrt{\sum_{k=1}^{n}(x_{ki} - x_{kj})^2}}, \tag{10}$$

where $i \neq j$ and s_{ij} denotes the similarity between the i-th and j-th features.

Then, based on the popular method K Nearest Neighbors (KNN), we select the top K similar features for each feature and set their weights as 1 and the weights of the remaining features as 0, respectively, *e.g.*,

$$s_{ij} = \begin{cases} 1 \text{ if } s_{ij} > \tilde{s}_{iK} \\ 0 \text{ otherwise,} \end{cases} \tag{11}$$

where $\tilde{s}_{iK}$ is the K-th largest similarity in the remaining $d-1$ features for the i-th feature. But this will generate an asymmetric matrix $\mathbf{S}$. To attain a symmetric adjacency matrix $\mathbf{S}$, we update it by $\mathbf{S} = (\mathbf{S} + \mathbf{S}^T)/2$.

Similar to the adjacency matrix $\mathbf{S}$, we can obtain $\mathbf{A}$ based on (10) and (11) by calculating the similarity among samples.

3) Comparison Methods: The comparative methods include 3 traditional machine learning methods and 7 deep learning methods, we list their details as follows:

- L_1-Norm Support Vector Machines(L_1**SVM**) [38] utilizes SVM to complete the classification task, while using L_1-norm to select important features.
- Sparse Learning Support Vector Machines (**SLSVM**) [17] adopts SVM to conduct the classification task, while using sparse learning to control the sparsity of the feature matrix and obtain important features.
- Random Forest (**RF**) [27] integrates multiple base learners (*e.g.*, decision tree) to form a strong classifier and ranks the importance of features during training.
- Adaptive boosting(**Adaboost**) [22] optimizes model training by continuously boosting the weights of misclassified samples.
- Graph Convolutional Network (**GCN**) [12] leverages both the feature information and structural information of nodes to obtain robust feature representations.
- Graph Attention Network (**GAT**) [30] employs multi-head attention mechanisms to assign link weights between nodes and gradually adjusts the adjacency matrix to the optimum.
- Attention-based Graph Neural Network(**AGNN**) [29] replaces the fully-connected layers of the GCN model with the propagation layers under attention mechanisms, so as to learn a dynamic and adaptive neighborhood.
- Approximate personalized propagation of neural predictions (**APPNP**) [13] combines GCN with PageRank to aggregate information from more distant neighbors without adding redundant graph convolutional parameters.
- Sample Reweight (**SR**) [21] utilizes the clean verification set to calculate the weight of each sample for reweighting all samples.
- Meta-Weight Net (**MWN**) [26] adopts a MLP to produce adaptive sample weights to weight the loss function as well as the clean verification set to guide model training.
- Interpretable Dynamic Graph Convolutional Networks (**IDGCN**) [40] integrates dynamic graph learning and graph convolution to learn the optimal graph structure and provides feature interpretability.

Traditional machine learning methods (*i.e.*, L_1SVM, RF and Adaboost) do not construct graphs to leverage the structural information among samples or features. Meanwhile, for the deep learning methods, we consider structural information by constructing the adjacency matrix into GCNs. Additionally, all deep learning methods are trained in a supervised manner. Moreover, we compare SFNet with L_1SVM, RF and IDGCN on feature interpretability. Similarly, we compare SFNet with Adaboost, SR and MWN on sample interpretability.

4) Experimental Setting: We conducted our experiments by using the framework PyTorch on a server with 8 NVIDIA GeForce 3090 (24 GB memory each).

To better compare the aforementioned methods with the proposed FSNet on a small number of sample data, we adopted the five-fold cross-validation and independently conducted cross-validation experiments 20 times for all methods on four datasets. Finally, we reported their average results.

We evaluated all methods by using four popular metrics, including classification accuracy, specificity, sensitivity and the AUC score. Additionally, in AD diagnosis, besides the four evaluation metrics mentioned above, we also presented interpretability results on selecting significant features (brain regions) and samples.

5) Implemental Details: We adjusted hyper-parameters for each method by referring to the corresponding literature to output their best results. For our method, we set the maximum epochs as 500, the learning rate as 0.005, and two graph convolutional layers, *i.e.* $L = 2$ in (1), (5) and (7). We also set $K_f = d \times \lambda_f$, where d is the number of features, $\lambda_f \in \{0.1, 0.2, \ldots, 1\}$ and $K_s = n \times \lambda_s$, where n is the number of samples, $\lambda_s \in \{0.1, 0.2, \ldots, 1\}$ is utilized to obtain the best classification performance on four datasets, respectively. And we also list optimal λ_f and λ_s on four datasets in the appendix section. Based on feature weight vector $\mathbf{w}_f$ and sample weight vector $\mathbf{w}_s$, we regard the features and samples with weights larger than 0 as significant ones, and the other with weights of 0 are trivial features or samples.

B. Results and Analysis

1) Classification Results: Tables II and III present the classification performance of all methods on four datasets, where we bold the best result of each metric in this dataset. They illustrate that: 1) Deep learning methods usually obtain superior performance over the traditional machine learning methods on the four datasets in most of cases. 2) The proposed FSNet can obtain the best results, including accuracy (ACC), sensitivity (SEN), specificity (SPE) and AUC scores, among all methods on the four datasets in almost all cases. Specifically, FSNet obtains slightly higher results than comparative methods on AD-NC and AD-MCI datasets in term of the four metrics except SEN on AD-MCI, and it achieves significantly better results than the best

TABLE II
THE CLASSIFICATION PERFORMANCE ON AD-NC AND AD-MCI DATASETS. WE BOLD THE BEST RESULT IN EACH SETTING

	AD-NC				AD-MCI			
Method	ACC	SEN	SPE	AUC	ACC	SEN	SPE	AUC
L_1SVM	76.2 ± 0.55	76.9 ± 0.61	73.8 ± 0.51	74.9 ± 0.51	66.9 ± 0.86	69.7 ± 0.93	65.6 ± 0.59	68.2 ± 0.87
SLSVM	76.4 ± 0.58	78.6 ± 0.67	74.4 ± 0.52	76.5 ± 0.62	68.2 ± 0.92	70.2 ± 0.88	64.9 ± 0.94	68.4 ± 0.87
RF	78.5 ± 0.51	81.6 ± 0.60	72.6 ± 0.43	77.1 ± 0.77	68.5 ± 0.94	70.1 ± 0.76	66.4 ± 0.87	68.4 ± 0.81
Adaboost	77.4 ± 0.54	79.6 ± 0.63	74.8 ± 0.52	76.9 ± 0.59	67.4 ± 0.82	69.9 ± 0.68	65.7 ± 0.64	67.3 ± 0.64
GCN	82.8 ± 0.51	81.6 ± 0.59	83.4 ± 0.34	82.4 ± 0.77	68.9 ± 0.78	70.5 ± 0.64	67.8 ± 0.37	68.7 ± 0.63
GAT	82.9 ± 0.46	82.0 ± 0.41	83.7 ± 0.45	83.1 ± 0.53	69.8 ± 0.52	71.6 ± 0.72	67.8 ± 0.48	68.2 ± 0.67
AGNN	83.0 ± 0.54	81.7 ± 0.52	83.7 ± 0.37	83.3 ± 0.86	70.2 ± 0.64	70.6 ± 0.62	69.5 ± 0.56	70.0 ± 0.72
APPNP	81.9 ± 0.58	80.8 ± 0.62	82.3 ± 0.53	81.1 ± 0.81	69.9 ± 0.60	70.6 ± 0.46	68.4 ± 0.46	68.9 ± 0.57
SR	82.1 ± 0.41	81.0 ± 0.59	82.9 ± 0.48	82.5 ± 0.70	72.0 ± 0.58	73.1 ± 0.67	71.5 ± 0.41	72.2 ± 0.71
MWN	83.4 ± 0.41	82.2 ± 0.52	84.1 ± 0.42	83.6 ± 0.60	72.6 ± 0.59	**75.2 ± 0.64**	71.9 ± 0.49	72.3 ± 0.46
IDGCN	83.3 ± 0.47	82.4 ± 0.48	84.5 ± 0.35	83.5 ± 0.74	71.8 ± 0.73	72.5 ± 0.59	70.3 ± 0.66	71.4 ± 0.62
FSNet	**84.4 ± 0.36**	**83.6 ± 0.44**	**85.9 ± 0.42**	**84.3 ± 0.59**	**73.6 ± 0.56**	74.4 ± 0.56	**72.5 ± 0.41**	**74.9 ± 0.59**

TABLE III
THE CLASSIFICATION PERFORMANCE ON NC-MCI AND MCIN-MCIP DATASETS. WE BOLD THE BEST RESULT IN EACH SETTING

	NC-MCI				MCIn-MCIp			
Method	ACC	SEN	SPE	AUC	ACC	SEN	SPE	AUC
L_1SVM	63.5 ± 0.70	57.3 ± 0.92	66.3 ± 1.08	63.9 ± 0.62	64.4 ± 0.57	66.5 ± 0.70	63.0 ± 0.68	64.3 ± 0.94
SLSVM	63.8 ± 0.72	58.7 ± 0.83	67.2 ± 0.54	64.2 ± 0.78	65.6 ± 0.66	67.2 ± 0.72	63.2 ± 0.72	66.2 ± 0.64
RF	65.5 ± 0.89	61.8 ± 0.81	67.1 ± 0.94	64.5 ± 0.68	65.8 ± 0.58	66.1 ± 0.78	63.9 ± 0.88	64.8 ± 0.67
Adaboost	64.9 ± 0.72	60.7 ± 0.78	65.9 ± 0.81	64.1 ± 0.69	65.2 ± 0.70	67.0 ± 0.81	63.7 ± 0.65	64.1 ± 0.59
GCN	65.4 ± 0.64	62.3 ± 0.82	69.6 ± 0.76	65.0 ± 0.71	66.6 ± 0.42	69.1 ± 0.58	62.6 ± 0.75	65.8 ± 0.83
GAT	65.6 ± 0.85	61.3 ± 0.74	68.3 ± 0.61	64.4 ± 0.74	65.7 ± 0.51	68.9 ± 0.70	62.5 ± 0.64	66.7 ± 0.72
AGNN	65.9 ± 0.65	63.2 ± 0.86	70.2 ± 0.53	65.9 ± 0.86	65.5 ± 0.46	67.6 ± 0.71	63.3 ± 0.63	65.5 ± 0.58
APPNP	66.0 ± 0.55	62.4 ± 0.46	66.9 ± 0.68	66.9 ± 0.72	65.2 ± 0.42	66.7 ± 0.83	65.3 ± 0.52	67.3 ± 0.64
SR	67.5 ± 0.54	65.3 ± 0.66	68.4 ± 0.46	67.1 ± 0.68	66.7 ± 0.34	69.2 ± 0.56	63.4 ± 0.48	66.2 ± 0.45
MWN	66.8 ± 0.51	64.1 ± 0.62	69.2 ± 0.68	68.0 ± 0.59	67.3 ± 0.41	70.6 ± 0.62	65.6 ± 0.52	67.8 ± 0.46
IDGCN	67.2 ± 0.63	66.0 ± 0.82	69.0 ± 0.58	66.5 ± 0.75	66.9 ± 0.35	68.2 ± 0.61	64.5 ± 0.63	66.9 ± 0.66
FSNet	**71.8 ± 0.52**	**68.4 ± 0.61**	**73.7 ± 0.62**	**72.0 ± 0.64**	**70.2 ± 0.30**	**72.0 ± 0.58**	**67.4 ± 0.59**	**70.5 ± 0.37**

competitors on NC-MCI and MCIn-MCIp datasets. For instance, on NC-MCI, the results of FSNet is 4.3%, 2.4%, 3.5% and 4.0% over the best competitors in term of ACC, SEN, SPE and AUC, respectively.

The possible reasons for aforementioned two observations are: 1) Traditional machine learning methods cannot extract powerful discriminative feature representations compared to deep learning methods, and they also fail to leverage the correlated relationship among samples. Hence, they obtain inferior classification performance to deep learning methods, even though they can interpret the significance of features or samples. 2) Although the deep learning methods can utilize the adjacency matrix to extract powerful features, they either do not consider the significance of both samples and features or only take into account one of them. By contrast, our method can simultaneously take into account the significance of both samples and features, and meanwhile employ them to guide model training for classification.

2) Feature Interpretability: To evaluate the feature interpretability of the proposed FSNet, we compare it with three feature interpretation methods, such as L_1SVM, RF and IDGCN. Specifically, we implement the four methods 20 times using the five-fold cross-validation, thereby totally running feature selections 100 times on the four datasets, and each time records the 20 most significant features. Therefore, we can count the frequency of each feature during the 100 feature selections. Finally, the 20 features with the highest number of occurrences were selected as the significant features of the method.

Tables IV and V display the indexes of top 20 features and their orders of importance on the four datasets. Additionally, we show the name of brain regions corresponding to the features in the appendix. Moreover, we visualized the brain regions selected by the four methods on the AD-NC dataset in Fig. 3.

a) Ranking performance evaluation: The literature [2] demonstrates that the most associated brain regions with AD are hippocampal formation (30, 69) and amgydata (76, 83). Based on Tables IV and V, we can observe that all the four methods can select all or most of the four most important brain regions. However, the four methods obtain inconsistent rank orders of brain regions. To further compare the performance of FSNet with the others in terms of feature interpretability, we present their ranking performance evaluated by using the average precision (AP) [3], which is defined as:

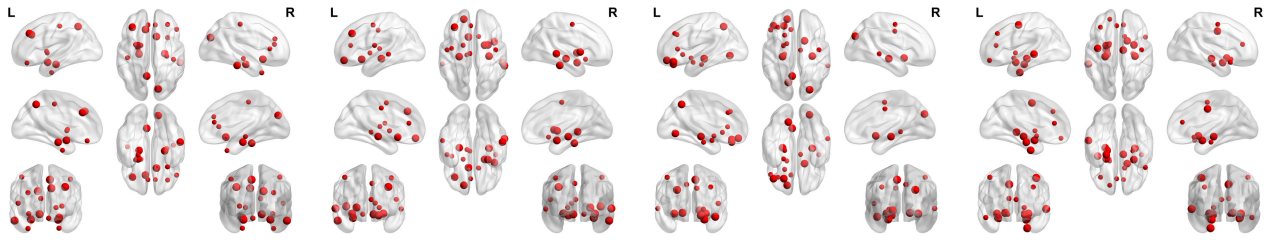
$$AP = \frac{\sum_{i=1}^{N_{total}} (P(i) * e(i))}{N_{imp}}, \tag{12}$$

TABLE IV
THE INDEXES OF TOP 20 IMPORTANT BRAIN REGIONS ON AD-NC AND AD-MCI. THE BOLD NUMBER DENOTES THE MOST RELEVANT BRAIN REGIONS FOR THE AD DISEASE FOUND BY EACH METHOD

Method	AD-NC	AD-MCI
L_1SVM	84,89,**76**,41,36,8,**78**,50,**17**,25,19,2,53,12,73,**30**,13,64,35,54	**30**,84,68,64,**5**,86,21,31,52,38,27,9,73,33,62,**69**,25,81,82,85
RF	**30**,8,3,11,**83**,**69**,27,52,84,80,**76**,12,46,36,24,20,**55**,48,**5**,70	8,**30**,**83**,**5**,11,85,**69**,80,47,4,**55**,**78**,7,39,17,31,36,41,28,67
IDGCN	21,27,32,49,**69**,60,**30**,58,41,**83**,8,50,**76**,80,93,51,14,**17**,22,**5**	24,70,52,**69**,84,22,20,23,12,11,**30**,**55**,**76**,8,36,46,64,**83**,80,**5**
FSNet	**69**,**17**,41,30,**76**,**83**,24,48,46,28,35,64,51,**78**,**5**,93,8,20,58,33	**17**,**30**,21,**76**,**55**,22,**69**,48,64,35,**83**,**78**,8,**5**,12,20,27,71,51,36

TABLE V
THE INDEXES OF TOP 20 IMPORTANT BRAIN REGIONS ON NC-MCI AND MCIN-MCIP. THE BOLD NUMBER DENOTES THE MOST RELEVANT BRAIN REGIONS FOR THE AD DISEASE FOUND BY EACH METHOD

Method	NC-MCI	MCIn-MCIp
L_1SVM	63,**30**,14,89,13,41,**83**,**78**,56,34,29,18,12,44,82,50,59,19,51,53	1,50,14,65,23,28,87,53,**76**,**30**,**83**,90,40,35,21,55,32,20,**17**
RF	**69**,3,8,32,**76**,85,11,**83**,92,46,**55**,64,25,36,20,27,40,68,1,65	**5**,36,33,**83**,1,**30**,46,**55**,39,**17**,11,80,48,86,42,13,27,90,**76**,89
IDGCN	41,22,**5**,44,80,**78**,14,**55**,**69**,89,50,**30**,20,34,**17**,**83**,29,13,10,63	21,**69**,**30**,22,64,**78**,62,70,52,48,**83**,2,86,33,81,4,68,73,**17**,10
FSNet	**69**,22,64,**55**,**17**,79,**76**,**68**,53,**30**,63,**83**,**5**,54,**78**,10,18,38,48,50	**30**,**78**,25,**69**,**83**,**17**,48,**5**,26,48,22,53,57,**76**,64,60,70,43,13,6



(a) L_1SVM (b) RF (c) IDGCN (d) FSNet

Fig. 3. Top 20 brain regions selected on the AD-NC dataset by four methods.

where N_{imp} and N_{total} represent the number of important and all brain regions in the sequence, respectively; $e(i) \in \{0, 1\}$ denotes whether the i-th brain region is important or not and $P(i)$ is the accuracy of the first i brain regions containing important ones, *i.e.*,

$$P(i) = \frac{m_i}{i}, \tag{13}$$

where m_i is the number of searched important brain regions in the first i brain regions.

In our experiments, we regard hippocampal formation (30, 69) and amgydata (76, 83) as important features. Then, we calculated the AP for each sequence in Tables IV and V based on (12) and (13), and displayed the results of the four methods in the first part of Table VI. As we can see, the AP of FSNet is higher than that of the best competitors by 0.127, 0.163 and 0.237 in AD-NC, AD-MCI and MCIn-MCIp, respectively. Only in the NC-MCI, the AP of RF is slightly higher (0.013) than that of FSNet. These results illustrate the effectiveness of our method, which has superior ranking performance over the others.

In addition, the precentral gyrus (5, 55) and parahippocampal gyrus (17, 78) are also considered to be closely related to AD in [16] and referred to them as sub-important brain regions. For these sub-important brain regions, FSNet could also obtain better selection and ranking performance. For clarity, we present their ranking performance in the middle part of Table VI.

TABLE VI
AP WITH THE IMPORTANT BRAIN REGIONS ON TOP 20 SELECTED FEATURES. WE BOLD THE BEST RESULT IN EACH DATASET

Method	AD-NC	AD-MCI	NC-MCI	MCIn-MCIp
	Search indexes: (30,69,76,83)			
L_1SVM	0.115	0.281	0.196	0.146
RF	0.565	0.315	**0.510**	0.285
IDGCN	0.273	0.321	0.280	0.359
FSNet	**0.692**	**0.484**	0.497	**0.596**
	Search indexes: (5,55,17,78)			
L_1SVM	0.090	0.050	0.030	0.040
RF	0.040	0.424	0.022	0.237
IDGCN	0.038	0.046	0.327	0.068
FSNet	**0.211**	**0.484**	**0.386**	**0.302**
	Search indexes: (30,69,76,83,5,55,17,78)			
L_1SVM	0.150	0.198	0.145	0.137
RF	0.359	0.551	0.267	0.414
IDGCN	0.209	0.196	0.343	0.287
FSNet	**0.660**	**0.746**	**0.593**	**0.704**

Based on Tables IV–V, and the first two parts of Table VI, we can find that our proposed FSNet can select and assign high weights to sub-important brain regions, while other feature

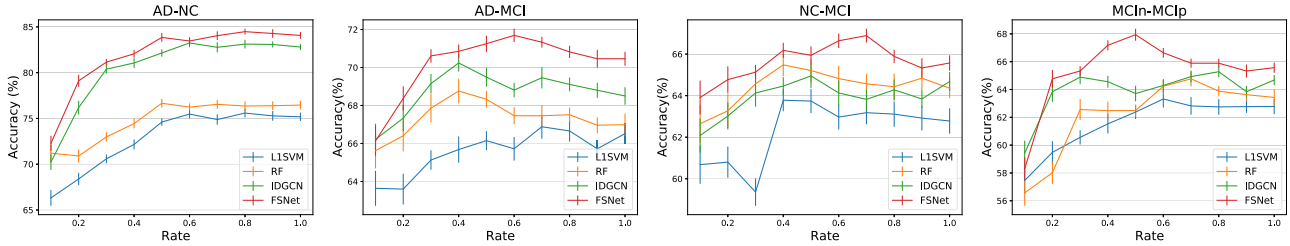


Fig. 4. The classification accuracy at different rates of features on four datasets.

selection methods are difficult to find or obtain good ranking performance. Therefore, when we regard the aforementioned eight features as important ones, the third part of Table VI shows that FSNet has a substantially better ranking performance over the others on all the four datasets. This also indicates the strength of our method on feature interpretation.

b) Classification performance evaluation: The evaluation of the important brain regions indexes referred to the priori medical knowledge. Next, we evaluate them by using the classification performance. Because a small number of high-dimensional data usually contain redundant or corrupted features. One of the most common strategies is to firstly perform feature selection and then only employ the selected features for model training again, since this strategy can reduce the complexity of model training and obtain superior classification performance with better interpretability [25], [37]. Based on this strategy, Fig. 4 shows the classification performance of the four methods on four datasets. As we can see, 1) All the four methods using part of full features can achieve or exceed the classification performance using full features on all four datasets. This suggests that there exist some redundant or corrupted features in the four datasets. For example, in the AD-NC dataset, FSNet using only 50% of the features can obtain a very similar accuracy to that using full features, and it obtains the best accuracy at using 80% of full features. 2) Although each method achieves the best accuracy of itself by using different numbers of features, the proposed FSNet consistently outperforms all comparative methods. This also infers the strength of our method in terms of feature interpretability.

Based on all the above analysis, we can conclude that: L_1SVM has poor classification and interpretability results; RF can obtain good interpretability results even though its classification performance is ordinary; IDGCN can obtain good classification results, but its interpretability results are relatively poor; only FSNet obtains the best interpretability results as well as classification performance.

3) Sample Interpretability: For sample interpretability, the most distinctive difference with feature is that we cannot know which samples are actually important based on a priori knowledge. Therefore, we cannot directly evaluate the sample interpretability only using the sample indexes and weights, which strategy is used for feature interpretability.

Therefore, we evaluate sample interpretability by using only the classification performance. We compare the proposed FSNet with three comparative methods, such as Adaboost, SR and MWN. Specifically, we first utilize the four methods to select significant samples and then employ them for model training again. Note that we found when the sample size was reduced to less than 30% of full samples in the experiments, the loss on the training set did not converge and continued to have large fluctuations, indicating training failed. Therefore, we only conducted experiments with sample sizes over 30%.

Fig. 5 presents the classification results on different selected rates of samples. We have the following observations: (1) All the four methods select only 70%-90% important samples involved in training can achieve the best classification performance of themselves on the three datasets, including AD-MCI, NC-MCI and MCIn-MCIp. However, in AD-NC, all the four methods need to employ all samples to obtain the highest accuracy. They suggest the three datasets, including AD-MCI, NC-MCI and MCIn-MCIp, contain corrupted samples. (2) FSNet obtains the best performance on sample interpretability. At different rates, our method can obtain higher accuracy than the others on all the four datasets. For example, FSNet achieves 5.1% higher accuracy than the best competitor, SR, on the NC-MCI dataset.

C. Ablation Analysis

To investigate the effects of feature and sample interpretability on model classification, we conduct experiments on four conditions: 1) Without interpreting the significance of both features and samples; 2) Only considering feature interpretability; 3) Only considering sample interpretability; and 4) Considering both feature and sample interpretability.

Fig. 6 shows the classification results of the proposed FSNet under the four conditions. It illustrates that 1) Interpreting the significance of features or samples can boost model classification performance and whether feature or sample is more important depending on the dataset; 2) Simultaneously interpreting the significance of features and samples can further boost model performance on all the four datasets, compared to that only interpreting one of them, which indicates that the two parts of our FSNet are important.

D. Hyper-Parameters Analysis

We investigate the influence of two hyper-parameters (λ_f and λ_s) in our FSNet. They control the sparsity of features and

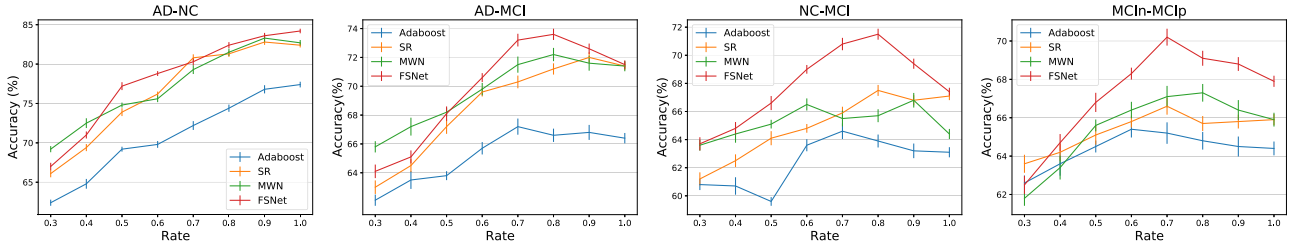


Fig. 5. The classification accuracy at different rates of samples on four datasets.

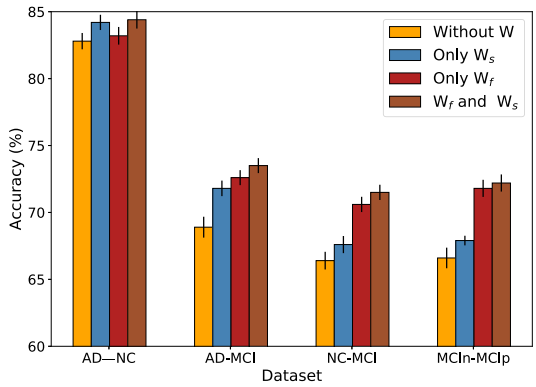
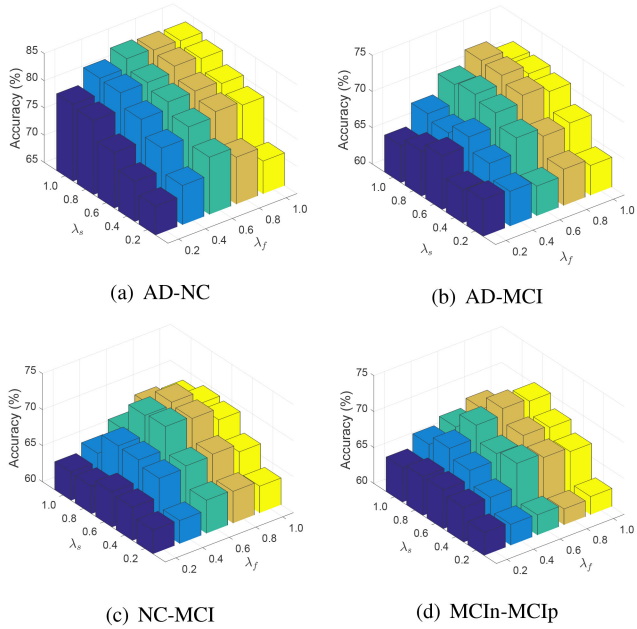


Fig. 6. Ablation study: the classification performance under four conditions.



(a) AD-NC (b) AD-MCI

(c) NC-MCI (d) MCIn-MCIp

Fig. 7. The classification accuracy of FSNet at different parameter settings on λ_f and λ_s.

samples in our FSNet, respectively. Specifically, we conduct experiments by ranging their values from 0.2 to 1 and visualize the classification results on all datasets in Fig. 7.

As shown in Fig. 7, the performance of FSNet is relatively poor when λ_f and λ_s are during the range of $[0.2, 0.6]$. However, it can achieve an optimal or near-optimal classification performance when selecting 80% of the features and samples from the four datasets. This suggests that FSNet is not sensitive to parameters and we can empirically set $\lambda_f = \lambda_s = 0.8$ in most cases.

E. Discussion

In this section, we present a comprehensive discussion of our FSNet in AD diagnosis in terms of classification performance, feature interpretability, and sample interpretability. For classification performance, the proposed FSNet has promising and superior performance over the others on all the four datasets, probably because we select and weight features and samples during model training. In terms of feature interpretability, we utilize both the priori medical knowledge and classification performance to illustrate that FSNet can achieve better feature selection performance than L_1SVM, RF and IDGCN, and feature selection can boost model performance, which suggests that there are existing redundant or corrupted features in the ADNI database. For sample interpretability, by continuously adding important samples to the model for classification, FSNet also has better or at least competitive performance to the others on sample interpretability. This is because FSNet can select and meanwhile weight the significant samples to avoid or mitigate the effects of noise. Therefore, FSNet can not only obtain excellent classification performance, but also attain promising feature and sample interpretability by selecting and weighting significant features and samples.

IV. Conclusion

In this paper, we propose a novel graph convolutional network, FSNet, for AD diagnosis with simultaneously obtaining feature and sample interpretability. To fulfill this goal, we employ two sub-networks by using the $L_{2,1}$-norm to select significant features and samples, respectively. Experimental results demonstrated the strength of the proposed network in terms of classification performance and interpretability, compared to state-of-the-art methods.

In the experiments, we only validate the proposed method by using the single-view data, which contains limited information, thereby possibly restricting its model performance. In the future, we will investigate its performance on multi-view data, such as PET, fMRI and sMRI.

APPENDIX

1	medial front-orbital gyrus right	47	middle occipital gyrus right
2	middle frontal gyrus right	48	middle temporal gyrus left
3	lateral ventricle left	49	lingual gyrus left
4	insula right	50	superior frontal gyrus left
5	precentral gyrus right	51	nucleus accumbens left
6	lateral front-orbital gyrus right	52	occipital lobe WM left
7	cingulate region right	53	postcentral gyrus left
8	lateral ventricle right	54	inferior frontal gyrus right
9	medial frontal gyrus left	55	precentral gyrus left
10	superior frontal gyrus right	56	temporal lobe WM left
11	globus palladus right	57	medial front-orbital gyrus left
12	globus palladus left	58	perirhinal cortex right
13	putamen left	59	superior parietal lobule right
14	inferior frontal gyrus left	60	lateral front-orbital gyrus left
15	putamen right	61	perirhinal cortex left
16	frontal lobe WM right	62	inferior temporal gyrus left
17	parahippocampal gyrus left	63	temporal pole left
18	angular gyrus right	64	entorhinal cortex left
19	temporal pole right	65	inferior occipital gyrus right
20	subthalamic nucleus right	66	superior occipital gyrus left
21	nucleus accumbens right	67	lateral occipitotemporal gyrus right
22	uncus right	68	entorhinal cortex right
23	cingulate region left	69	hippocampal formation left
24	fornix left	70	thalamus left
25	frontal lobe WM left	71	parietal lobe WM right
26	precuneus right	72	insula left
27	subthalamic nucleus left	73	postcentral gyrus right
28	posterior limb of internal capsule left	74	lingual gyrus right
29	posterior limb of internal capsule right	75	medial frontal gyrus right
30	hippocampal formation right	76	amygdala left
31	inferior occipital gyrus left	77	medial occipitotemporal gyrus left
32	superior occipital gyrus right	78	parahippocampal gyrus right
33	caudate nucleus left	79	anterior limb of internal capsule right
34	supramarginal gyrus left	80	middle temporal gyrus right
35	anterior limb of internal capsule left	81	occipital pole right
36	occipital lobe WM right	82	corpus callosum
37	middle frontal gyrus left	83	amygdala right
38	superior parietal lobule left	84	inferior temporal gyrus right
39	caudate nucleus right	85	superior temporal gyrus right
40	cuneus left	86	middle occipital gyrus left
41	precuneus left	87	angular gyrus left
42	parietal lobe WM left	88	medial occipitotemporal gyrus right
43	temporal lobe WM right	89	cuneus right
44	supramarginal gyrus right	90	lateral occipitotemporal gyrus left
45	superior temporal gyrus left	91	thalamus right
46	uncus left	92	occipital pole left
		93	fornix right

Fig. 8. The names of the selected brain regions in this work.

TABLE VII
OPTIMAL PARAMETERS ON FOUR BINARY DATASETS

Datasets	λ_f	λ_s	Datasets	λ_f	λ_s
AD-NC	0.8	1.0	AD-MCI	0.6	0.8
NC-MCI	0.7	0.8	MCIn-MCIp	0.5	0.7

REFERENCES

[1] S. Basaia *et al.*, "Alzheimer's disease neuroimaging initiative, *et al.* automated classification of alzheimer's disease and mild cognitive impairment using a single mri and deep neural networks," *NeuroImage: Clin.*, vol. 21, 2019, Art. no. 101645.

[2] M. F. Beal *et al.*, "Reduced numbers of somatostatin receptors in the cerebral cortex in alzheimer's disease," *Science*, vol. 229, no. 4710, pp. 289–291, 1985.

[3] A. Bellogín, P. Castells, and I. Cantador, "Statistical biases in information retrieval metrics for recommender systems," *Inf. Retrieval J.*, vol. 20, no. 6, pp. 606–634, 2017.

[4] H. Dai, A. Nazi, Y. Li, B. Dai, and D. Schuurmans, "Scalable Deep Generative Modeling for Sparse Graphs," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 2302–2312.

[5] A. Farooq, S. Anwar, M. Awais, and S. Rehman, "A deep CNN based multi-class classification of Alzheimer's disease using MRI," in *Proc. IEEE Int. Conf. Imag. Syst. Techn.*, 2017, pp. 1–6.

[6] J. Gan *et al.*, "Multi-graph fusion for dynamic graph convolutional network," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, 2022.

[7] J. Gan, Z. Peng, X. Zhu, R. Hu, J. Ma, and G. Wu, "Brain functional connectivity analysis based on multi-graph fusion," *Med. Image Anal.*, vol. 71, 2021, Art. no. 102057.

[8] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.

[9] R. Hu *et al.*, "Multi-band brain network analysis for functional neuroimaging biomarker identification," *IEEE Trans. Med. Imag.*, vol. 40, no. 12, pp. 3843–3855, Dec. 2021.

[10] B. Jie, D. Zhang, B. Cheng, and D. Shen, "Manifold regularized multi-task feature selection for multi-modality classification in alzheimer's disease," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2013, pp. 275–283.

[11] N. J. Kabani, D. MacDonald, C. J. Holmes, and A. C. Evans, "3D anatomical atlas of the human brain," *NeuroImage*, vol. 7, p. 2, 1998.

[12] N. Thomas Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *J. Int. Conf. Learn. Representations (ICLR)*, 2016.

[13] J. Klicpera, A. Bojchevski, and S. GÜnnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," in *J. Int. Conf. Learn. Representations (ICLR)*, 2018.

[14] L. Li *et al.*, "Robust adaptive semi-supervised classification method based on dynamic graph and self-paced learning," *Inf. Process. Manage.*, vol. 58, no. 1, 2021, Art. no. 102433.

[15] X. Li *et al.*, "Braingnn: Interpretable brain graph neural network for fMRI analysis," *Med. Image Anal.*, vol. 74, 2021, Art. no. 102233.

[16] F. Liu, C.-Y. Wee, H. Chen, and D. Shen, "Inter-modality relationship constrained multi-modality multi-task feature selection for alzheimer's disease and mild cognitive impairment identification," *NeuroImage*, vol. 84, pp. 466–475, 2014.

[17] M. Liu, D. Zhang, E. Adeli, and D. Shen, "Inherent structure-based multiview learning with multitemplate feature representation for alzheimer's disease diagnosis," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 7, pp. 1473–1482, Jul. 2015.

[18] K. Mortensen and T. L. Hughes, "Comparing amazon's mechanical turk platform to conventional data collection methods in the health and medical research literature," *J. Gen. Intern. Med.*, vol. 33, no. 4, pp. 533–538, 2018.

[19] S. Parisot *et al.*, "Disease prediction using graph convolutional networks: Application to autism spectrum disorder and alzheimer's disease," *Med. image Anal.*, vol. 48, pp. 117–130, 2018.

[20] L. Peng *et al.*, "Reverse graph learning for graph neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[21] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4334–4343.

[22] Robert E. Schapire, "Explaining adaboost," in *Empirical Inference*, Berlin, Germany: Springer, 2013, pp. 37–52.

[23] H. T. Shen *et al.*, "Heterogeneous data fusion for predicting mild cognitive impairment conversion," *Inf. Fusion*, vol. 66, pp. 54–63, 2021.

[24] X. Shi, F. Xing, Y. Xie, Z. Zhang, L. Cui, and L. Yang, "Loss-based attention for deep multiple instance learning," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 5742–5749.

[25] X. Shi *et al.*, "Loss-based attention for interpreting image-level prediction of convolutional neural networks," *IEEE Trans. Image Process.*, vol. 30, pp. 1662–1675, 2021.

[26] J. Shu *et al.*, "Meta-weight-net: Learning an explicit mapping for sample weighting," *Adv. Neural Inf. Process. Syst.*, pp. 32, 2019.

[27] V. Svetnik, A. Liaw, C. T., J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: A classification and regression tool for compound classification and QSAR modeling," *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 6, pp. 1947–1958, 2003.

[28] J. Tang, S. Alelyani, and H. Liu, "Feature Selection for Classification: A Review," *Data Classification: Algorithms Appl.*, pp. 37–64, 2014.

[29] K. K Thekumparampil, C. Wang, S. Oh, and L.-J. Li, "Attention-based graph neural network for semi-supervised learning," in *J. Int. Conf. Learn. Representations (ICLR)*, 2018.

[30] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *Statistics*, vol. 1050, pp. 20, 2017.

[31] C. Wang, B. Samari, and K. Siddiqi, "Local spectral graph convolution for point set feature learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 52–66.
[32] Q. Wang, Y. Zhou, W. Zhang, Z. Tang, and X. Chen, "Adaptive sampling using self-paced learning for imbalanced cancer data pre-diagnosis," *Expert Syst. Appl.*, vol. 152, 2020, Art. no. 113334.
[33] M. N. Wernick, Y. Yang, J. G. Brankov, G. Yourganov, and S. C. Strother, "Machine learning in medical imaging," *IEEE Signal Process. Mag.*, vol. 27, no. 4, pp. 25–38, Jul. 2010.
[34] Y. Yang, H. T. Shen, Z. Z. Ma Huang, and X. Zhou, "L2, 1-norm regularized discriminative feature selection for unsupervised," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1589–1594.
[35] C. Yuan, Z. Zhong, C. Lei, X. Zhu, and R. Hu, "Adaptive reverse graph learning for robust subspace learning," *Inf. Process. Manage.*, vol. 58, no. 6, 2021, Art. no. 102733.
[36] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, "On explainability of graph neural networks via subgraph explorations," in *Proc. Mach. Learn. Res.*, 2021, pp. 12241–12252.
[37] R. Zhao, Y. Hu, J. Dotzel, C. De Sa, and Z. Zhang, "Improving neural network quantization without retraining using outlier channel splitting," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7543–7552.
[38] J. Zhu, S. Rosset, R. Tibshirani, and Trevor J. Hastie, "1-norm support vector machines," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2003, pp. 49–56.
[39] X. Zhu, S. Zhang, Y. Zhu, P. Zhu, and Y. Gao, "Unsupervised spectral feature selection with dynamic hyper-graph learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 3016–3028, Jun. 2022, doi: 10.1109/TKDE.2020.3017250.
[40] Y. Zhu, J. Ma, C. Yuan, and X. Zhu, "Interpretable learning based dynamic graph convolutional networks for alzheimer's disease analysis," *Inf. Fusion*, vol. 77, pp. 53–61, 2022.