



Research article

An Alzheimer’s Disease classification network based on MRI utilizing diffusion maps for multi-scale feature fusion in graph convolution

Zhi Yang, Kang Li*, Haitao Gan*, Zhongwei Huang, Ming Shi and Ran Zhou

School of Computer Science, Hubei University of Technology, Wuhan 430068, China

* **Correspondence:** Email: 102111113@hbut.edu.cn, htgan01@hbut.edu.cn.

Abstract: Graph convolutional networks (GCN) have been widely utilized in Alzheimer’s disease (AD) classification research due to its ability to automatically learn robust and powerful feature representations. Inter-patient relationships are effectively captured by constructing patients magnetic resonance imaging (MRI) data as graph data, where nodes represent individuals and edges denote the relationships between them. However, the performance of GCNs might be constrained by the construction of the graph adjacency matrix, thereby leading to learned features potentially overlooking intrinsic correlations among patients, which ultimately causes inaccurate disease classifications. To address this issue, we propose an Alzheimer’s disease Classification network based on MRI utilizing diffusion maps for multi-scale feature fusion in graph convolution. This method aims to tackle the problem of features neglecting intrinsic relationships among patients while integrating features from diffusion mapping with different neighbor counts to better represent patients and achieve an accurate AD classification. Initially, the diffusion maps method conducts diffusion information in the feature space, thus breaking free from the constraints of diffusion based on the adjacency matrix. Subsequently, the diffusion features with different neighbor counts are merged, and a self-attention mechanism is employed to adaptively adjust the weights of diffusion features at different scales, thereby comprehensively and accurately capturing patient characteristics. Finally, metric learning techniques enhance the similarity of node features within the same category in the graph structure and bring node features of different categories more distant from each other. This study aims to enhance the classification accuracy of AD, by providing an effective tool for early diagnosis and intervention. It offers valuable information for clinical decisions and personalized treatment. Experimentation on the publicly accessible Alzheimer’s disease neuroimaging initiative (ADNI) dataset validated our method’s competitive performance across various AD-related classification tasks. Compared to existing methodologies, our approach captures patient characteristics more effectively and demonstrates superior generalization capabilities.

Keywords: Alzheimer’s disease; graph convolutional network; deep learning; diffusion maps; metric learning

1. Introduction

Alzheimer's disease (AD) is a severe neurodegenerative brain disorder with a wide-reaching global impact, affecting a substantial population of individuals [1, 2]. As of the present moment, approximately 90 million people have received a confirmed diagnosis of AD; according to projections [3, 4], this number is expected to escalate to 300 million by the year 2050. The early symptoms of AD primarily involves memory loss; as the condition progresses, patients encounter challenges in communication, spatial perception, and the control of bodily functions. This ailment stands as a predominant cause of mortality among the elderly [5, 6]. While a definitive cure for AD remains elusive, numerous therapeutic avenues exist, including pharmacological treatments, exercise regimens, and cognitive training, which are capable of mitigating symptoms and retarding the deterioration of the disease [7]. Therefore, the significance of accurately diagnosing AD is self-evident, particularly during the early stages, such as mild cognitive impairment (MCI), which marks the transitional phase from cognitive normal (CN) to AD [8]. An early diagnosis of AD can contribute to postponing the progression of the disease and enhancing the overall health condition of patients [9].

In recent decades, neuroimaging techniques have made significant strides in the field of early diagnosis of AD [10–12]. Among these techniques, magnetic resonance imaging (MRI) stands out as a non-invasive imaging modality. Its fundamental principle relies on the differential attenuation of energy in different structural environments within substances. By applying external gradient magnetic fields to detect emitted electromagnetic waves, MRI provides information about the positions and types of atomic nuclei within an object, subsequently generating images of the object's internal structure. Utilizing MRI technology, we can acquire detailed three-dimensional (3D) anatomical brain images that relatively and accurately portray structural changes in the brain affected by AD. MRI plays a pivotal role in clinical diagnoses and AD research [13, 14]. MRI scans offer high-contrast soft tissue images and exceptional spatial resolution, thus facilitating the observation of minute details and changes in brain tissue. This, in turn, aids in the classification of AD. Utilizing advanced MRI imaging techniques, researchers have devised various computational approaches, to assist in the classification of both AD and MCI.

In recent years, deep learning methods have made significant strides across various fields, including computer vision tasks in medical imaging classification. Convolutional neural networks (CNNs) have demonstrated remarkable capabilities in extracting features from MRI data, thereby considerably improving the diagnostic performance for AD [15, 16]. However, within the medical domain, acquiring large-scale, accurately labeled, and clean-feature data presents a significant challenge due to the high cost of data acquisition and the diverse clinical settings where subjective assessments by medical professionals prevail. These challenges can impose limitations on the effectiveness of deep learning methods [17]. Consequently, achieving robust results with deep learning approaches using a limited number of medical training samples has become exceedingly challenging. In the realm of AD diagnosis research and analysis, Graph convolutional networks (GCNs) have emerged as widely adopted deep learning frameworks. This is because GCNs can simultaneously consider semantic and structural information, thereby yielding more accurate

classification results compared to traditional machine learning methods, especially when the training samples are scarce [18, 19]. However, GCN's convolutional operations primarily rely on the Laplacian matrix of the graph, and are thereby constrained by the graph's own adjacency matrix.

In certain industrial applications, such as the case of a rolling bearing fault diagnosis, the incorporation of tangible physical information into the network architecture has been instrumental in enhancing the diagnostic accuracy [20]. Similarly, in collaborative fault identification for rotating machinery, employing an intermediate fusion strategy to amalgamate diverse modal features has contributed to an improved diagnostic precision [21]. Both these methodologies address issues by introducing missing feature information into the original framework. Similar to these approaches, in order to alleviate GCN's reliance on adjacency matrices, the recently emerged hybrid diffusion framework introduces diffusion features with low-dimensional embeddings through unsupervised manifold learning, thus improving the effectiveness of the model training [22]. While the hybrid diffusion framework aims to overcome limitations in information diffusion, there might be a potential concern: it could overlook feature information across different scales within the hybrid diffusion and fail to take the diverse neighborhood relationships into account.

To address the issue of missing feature information at different scales, we propose the multi-scale feature fusion graph convolutional network (MFF-GCN), drawing inspiration from recent developments in graph embedding based on random walks [23–25] and multi-scale graph convolution in a semi-supervised node classification [26]. We employ multiple GCN modules to perform convolution embedding on node features obtained from the diffusion of feature space node information at different neighbor counts. Subsequently, all output vectors from the GCN modules are integrated into a single classification network using a fully connected network (FCN) for the node classification of AD patients. This multi-scale feature fusion approach enables a more comprehensive consideration of feature information at different scales, ultimately enhancing the model's performance. Compared to certain current GCN-based modeling methods such as DAGCN [27], which considers the topological structure between variables and layer-wise characteristics to accommodate non-linearity between batches, and the proposal of a GCN-based soft sensor that applies graph concepts in process industries to capture unique local spatio-temporal correlations among variables [28]; both of these GCN-based modeling methods integrate distinct features between variables in their design to enhance model performance. However, in these two methods, the adjacency matrices are learned by the model, and there is no propagation of information among the neighborhood relationships in the feature space. In contrast, our MFF-GCN leverages unsupervised manifold learning to achieve propagation of neighborhood relationships in the feature space.

Within the hybrid diffusion framework, when propagating feature space information among nodes, changes in the node feature information are not constrained by the original graph structure. Consequently, the relationships between nodes in feature space may no longer adhere to the original graph structure. Therefore, dynamically adjusting the graph's structure is crucial for improving the model classification performance. Past research has often overlooked the inherent relationship between node features and the graph structure. For instance, in social networks, people with similar interests are more likely to become friends. To effectively integrate node features with the graph structure, it is necessary to dynamically adjust the input graph structure during the operation of the GCN. Hence, we propose a metric learning approach to dynamically adjust the input structure of the graph within the GCN. This adjustment aims to align the node features at various scales with the

graph structure, thereby making node features within the same category more proximate and those from different categories more distant. This, in turn, enables the graph convolution to effectively embed more similar node features.

Thus, we present an AD classification network based on MRI that utilizes diffusion maps for MFF-GCN. Initially, we conduct feature extraction on MRI image data by segmenting it according to brain regions. The MRI features of each patient are treated as nodes, and we create edges in the graph by calculating similarities among the features of all patients. Subsequently, we conduct the feature information of diffusion maps with varying neighbor counts on the constructed data, thereby resulting in features at different scales. Next, the features at various scales are separately processed through multiple distinct GCN modules. During this process, metric learning is incorporated into the GCN convolution to adjust the input node features at various scales to align within the same category more closely and maintain greater separation across different category nodes. This enables graph convolution to embed more closely related node features. Finally, the output vectors from all GCN modules are fused using a FCN to consolidate multi-scale feature information. The FCN provides the final node classification. The method proposed in this study demonstrates an ability to accurately classify AD patients and enhance the precision of AD classification tasks. The aim of this study is to enhance the accuracy of AD classification, thereby offering an effective tool for early diagnosis and intervention, as well as to furnish valuable insights for clinical decision-making and personalized treatment. Our primary contributions in this research can be summarized as follows:

- We have introduced a multi-scale feature fusion approach. Initially, we produce diffusion maps with varying neighbor counts to obtain multi-scale node features. Then, we mitigate the issue of feature information loss within the diffusion map process by amalgamating node features across different scales.
- To endow our model with finer-grained category discrimination capabilities, we have innovatively incorporated metric learning into the GCN for AD classification. This adjustment ensures that the node features at various scales align more closely within the same category and maintain greater separation across different category nodes in accordance with the graph structure, thereby enhancing the model performance.
- We have evaluated the effectiveness of this method on the publicly available Alzheimer's disease neuroimaging initiative (ADNI) dataset; experimental results demonstrate that our proposed approach performs well across multiple AD-related classification tasks.

The remaining organization of this research is as follows: Section 2 introduces the related works; Section 3 provides a detailed description of the specific workflow of the proposed method; Section 4 presents information regarding the data used for experiments, the experimental settings, the experimental results, and the discussions related to methods; finally, in Section 5, a comprehensive summary and overview of the entire content are provided.

2. Related works

2.1. Graph convolutional network

The Graph neural network (GNN) is a machine learning model designed for handling graph data. Its key characteristic lies in its ability to effectively describe the irregular structure of graph data by

considering dependencies among data samples. As one of the most popular models today, GNNs are widely employed in the processing and analysis of graph-structured data. Among these models, the GCN is a deep learning model inspired by CNNs and specifically tailored for graph data processing. The core idea of a GCN is to extend the convolution operation to graph data, thereby enabling convolution operations to be performed on graphs [29]. This model incorporates spectral graph convolution theory and utilizes Fourier transforms and Taylor expansion formulas to enhance filters. Assuming there is a graph $G = (A, X)$ containing N nodes, where A represents the adjacency matrix composed of edge weights, and X represents the feature matrix of N nodes, the information propagation rule of a GCN can be expressed as follows:

$$H^{l+1} = \sigma(\hat{A}H^lW^l) \quad (2.1)$$

In the above formula, $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, H^l represents the feature matrix of all nodes in layer 1, W^l represents the trainable weight matrix of layer 1, and σ denotes the activation function. $\tilde{A} = A + I$, and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ applies graph convolution theory to the construction of A . This process, $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, first transforms \tilde{A} into the Fourier domain, then truncates it using Chebyshev polynomials, and finally converts it back to the original time domain. With the utilization of the aforementioned propagation rule, a common two-layer GCN structure can be represented as follows:

$$\text{GCN}_{2\text{-layer-special}}(\tilde{A}, X) = \text{softmax}(\text{ReLU}(\hat{A}XW^0)W^1). \quad (2.2)$$

2.2. Graph node classification task

The task of graph node classification, also known as semi-supervised node classification, entails the classification of each node in the graph data, thereby predicting its respective category. In the field of graph representation learning, we can employ two primary methods to address this issue. Initially, we can train a machine to learn model by learning node embeddings, and subsequently employ this model for node classification. This approach is relatively intuitive, as node embedding learning allows us to represent nodes in a low-dimensional space, thereby rendering them as suitable for an application in traditional machine learning methods. Alternatively, we can perceive the problem of node classification in graphs from the perspective of information propagation. Within a graph, nodes exhibit interrelatedness and diffusion, with connected nodes having a higher likelihood of belonging to the same category. Consequently, we can regard the task of node classification as a process of information dissemination within the graph, where the propagation of information influences the ultimate classification labels of nodes. Graph node classification finds extensive applications in real-life scenarios, thereby encompassing various tasks related to the graph data analysis, such as social network analysis [30], bioinformatics [31], and knowledge graphs [32]. These tasks often involve a multitude of nodes and intricate relationships. Graph node classification methods offer us a robust tool to gain a deeper understanding of and harness the patterns and structures present in these data, thus driving advancements in various fields of research and applications.

2.3. Metric learning

Metric learning is a significant subfield within the machine learning domain, which aims to learn a metric (or distance) function to measure either the similarity or dissimilarity between data samples.

The fundamental idea behind metric learning is that by acquiring an appropriate metric function, similar samples can be mapped closer together, while dissimilar samples can be mapped farther apart, thereby enhancing the capture of the intrinsic structure and relationships within the data space. Metric learning assumes a pivotal role in the context of multi-scale feature fusion within GCNs. To effectively integrate node features with the original graph structure, there is a need to dynamically adjust the input graph structure during the process of GCNs. In the domains of classification and clustering, each dataset poses specific challenges. The distance metrics often lack effective learning capabilities, thereby resulting in unreliable outcomes in data classification. Therefore, a robust distance metric is essential for achieving reliable outcomes for the input data [33, 34]. Currently, there is research introducing the concept of metric learning within GCNs to enhance the feature representation of graph nodes [35, 36].

2.4. Alzheimer's disease classification with graph convolutional networks

In AD classification research involving GCNs, the initial adoption of GCN integrated both image and non-image features for predicting AD progression [37]. They employed a semi-supervised learning approach, and trained the GCNs model using a subset of labeled nodes to predict unlabeled nodes. Subsequently, several methods utilizing GCNs for AD classification have emerged. One approach involved combining various clinical features into multiple graphs and consolidating the classification results of each graph for early an AD diagnosis [38, 39]. Additionally, an initial module was introduced to capture the structural heterogeneity within and between graphs for predicting MCI conversion [40]. On another front, the fusion of GCNs with recurrent neural networks was explored to handle missing values while simultaneously predicting MCI-AD conversion [41]. However, these methods did not account for the diffusion of node information within the feature space. Furthermore, they did not find a more suitable distance metric to describe the adjacency matrix in the graph convolution process through metric learning.

3. Methods

In this section, we will introduce the proposed multi-scale feature fusion GCN for AD diagnoses based on MRI data, as shown in Figure 1. First, we perform a feature extraction based on brain regions from each subject's MRI images. We consider each patient's MRI features as nodes and establish a graph structure based on the similarity between nodes. Next, we apply two different diffusion map operations to the graph structure data, thereby allowing the diffusion of node information from the perspective of proximity in feature space to obtain multi-scale node features. Subsequently, we employ metric learning on the multi-scale node features obtained from the diffusion maps to make them adhere to a graph structure, where nodes of the same category are closer and nodes of different categories are farther apart. Then, graph convolution operations are applied to embed these multi-scale node features, thus resulting in the output of GCN modules for each scale. Finally, we merge the outputs of the two different scale GCN modules and the original graph structure data into a classification network for AD classification, thereby achieving the goal of multi-scale feature fusion. Additionally, in this paper, $X = [x_1, x_2, \dots, x_n] \in R^{N \times d}$ represents a matrix with N participants, where $x_i \in R^d (1 \leq i \leq N)$ denotes the d -dimensional features of the i -th participant.

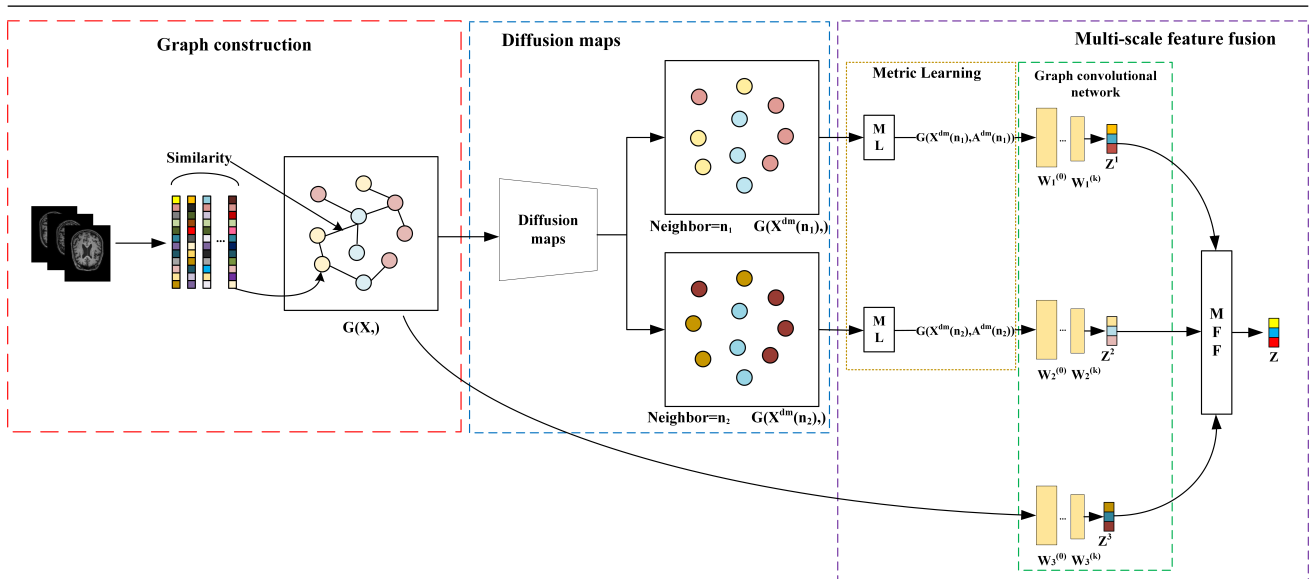


Figure 1. Architecture of the proposed MFF-GCN.

3.1. Graph construction

First, through preprocessing of MRI image data, we obtain brain region features for each subject as $x \in R^d$. We construct a graph where each node represents a subject, and we calculate the Euclidean distance between each pair of nodes x_i and x_j to create a similarity matrix, as shown in the following formula:

$$S(\rho(\cdot, \cdot)) = \exp\left(-\frac{\rho(\cdot, \cdot)^2}{2\sigma^2}\right) \tag{3.1}$$

Here, $S(\rho(\cdot, \cdot))$ is the Gaussian kernel function, $\rho(\cdot, \cdot)$ is the distance function, and σ is the kernel width. A larger variance σ leads to weaker connectivity between data nodes.

Next, we normalize and symmetrize the similarity matrix to obtain the adjacency matrix A . Then, we binarize A using the following formula:

$$A_{ij} = \begin{cases} 1 & A_{ij} \geq \mu \\ 0 & A_{ij} < \mu \end{cases} \tag{3.2}$$

where μ is the threshold for binarizing A . This process results in our graph structure data $G(X, A)$.

3.2. Diffusion maps

Diffusion maps are a spectral manifold learning method. They define a Markov random walk on a graph constructed from data points. After a random walk with a certain number of time steps, a diffusion map computes a distance function capturing the proximity between any two data points, also referred to as the diffusion distance. The goal is to obtain a low-dimensional manifold structure of the data while preserving the diffusion distances as much as possible.

For a given node feature X , the first step is to construct an affinity matrix C , from the given node feature X . Diffusion maps leverage the relationship between heat kernel diffusion and the Markov chain of random walks. The connectivity between two nodes can be defined as the probability of a

random walk transitioning from one node to another. Typically, a Gaussian kernel function is chosen to define the connectivity between nodes, which is calculated as follows:

$$\rho(x_i, x_j) = \|x_i - x_j\|_2, \quad x_i, x_j \in X \quad (3.3)$$

Through the given kernel function $S(\rho(x_i, x_j))$, we can obtain the affinity matrix C constructed from the node feature X as follows:

$$C_{ij} = S(\rho(x_i, x_j)), \quad x_i, x_j \in X \quad (3.4)$$

Then, we normalize the affinity matrix C so that the sum of each row equals 1. This results in a normalized matrix $P = P^{(1)} = \{p_{ij}^{(1)}\}$, where the elements are defined as following: $p_{ij}^{(1)} = \frac{c_{ij}}{\sum_n c_{in}}$. This normalized matrix P can be interpreted as a Markov state transition matrix on the dataset X . It represents the probabilities of random walks between various nodes. Therefore, P indicates the probability of moving from one point to another after one step of random walking. If we perform t steps of random walking, the corresponding transition probability is denoted as $P^t = (P)^t$. In this way, the state transition matrix P , after t steps of random walking, yields the diffusion mapping matrix $D = P^t$.

The diffusion mapping matrix D can be subjected to eigenvalue decomposition to obtain k largest eigenvalues and their corresponding eigenvectors. From this, we can derive as following: $\psi(x)D = \lambda\psi(x)$. This reveals that the corresponding eigenvectors can represent a new set of coordinates in the feature space for the dataset. We obtain the diffusion mapping into a k -dimensional space as follows:

$$X^{dm} = \Psi^{(t)}(x) = \sum_k \lambda_k^t \psi_k(x), \quad x \in X \quad (3.5)$$

Here, t represents the number of diffusion iterations, λ_k represents the k -th eigenvalue of the node feature x , and $\psi_k(x)$ represents the k -th eigenvector of the node feature x . $\Psi^{(t)}(x)$ represents the k -dimensional mapping of node features after t iterations of the diffusion map, also denoted as X^{dm} .

3.3. Adaptive adjacency matrix based on metric learning

After the diffusion map, the diffusion features X^{dm} exhibit changes in their feature information, which are no longer constrained by the original graph structure. To effectively embed node features through the graph structure, there is a need to dynamically adjust the input graph structure during the GCN process. Thus, we propose a metric learning approach to dynamically adjust the graph's input structure within the GCN process, thereby ensuring that node features from various scales align more closely for nodes of the same category while being distant for nodes of different categories, which ultimately enhances the performance of GCN.

We employ a metric learning approach to adaptively update the graph structure based on input feature [36]. This involves learning a metric function for pairwise similarity represented by a learned metric matrix over the input features. We define a non-negative function between node features x_i and x_j as follows: $\rho_M : X \times X \rightarrow R^+$,

$$\rho_M(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)} \quad (3.6)$$

If $M = I$, Eq (3.6) simplifies to the Euclidean distance. In our model, $M = W^d (W^d)^T$ is a symmetric positive semi-definite matrix, where W_d is the trainable weight of the metric learning. $W^d \in R^{d \times d}$ represents a transformation basis for measuring the Euclidean distance between x_i and x_j . Then, we compute the metric learning-based adjacency matrix A_{ij}^{dm} using Eq (3.1):

$$A_{ij}^{dm} = S(\rho_M(x_i, x_j)), \quad x_i, x_j \in X^{dm} \quad (3.7)$$

The matrix W_d is optimized during the subsequent training of the GCN network, thus resulting in A^{dm} as the adjacency matrix for the node features.

3.4. Multi-scale feature fusion

In a diffusion map, the choice of the neighbor count, denoted as n is a crucial parameter. It determines how many near neighboring data points are considered when constructing the similarity matrix for the data. The neighbor count influences the sparsity of the similarity matrix and the connectivity between data points. Opting for a smaller neighbor count captures local data structures, while choosing a larger neighbor count captures global data structures. Therefore, the neighbor count allows you to control the extent of diffusion and the structure of the embedding space. Selecting different neighbor counts can result in different embedding outcomes, which can be reflected in the scale of node features. Thus, when using different neighbor counts, one can obtain diffusion node features $X^{dm}(n)$ at different scales, where n represents the neighbor count used in diffusion maps. By applying Eqs (3.6) and (3.7), one can derive the corresponding $A^{dm}(n)$ for these diffusion node features, thereby obtaining different-scale graph structural data $G(X^{dm}(n), A^{dm}(n))$.

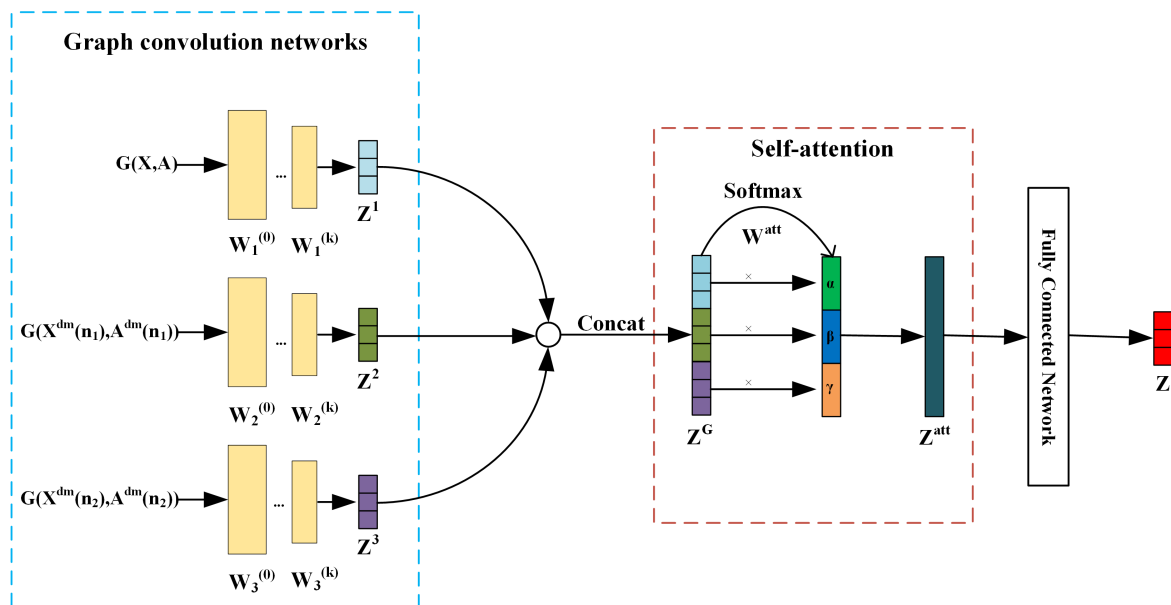


Figure 2. Multi-scale feature fusion network structure.

As illustrated in Figure 2, we consider three instances of GCN, namely $\{GCN(X^{dm}(n_1), A^{dm}(n_1)), GCN(X^{dm}(n_2), A^{dm}(n_2)), GCN(X, A)\}$, $G(X^{dm}(n_1), A^{dm}(n_1))$, $G(X^{dm}(n_2), A^{dm}(n_2))$ and $G(X, A)$, which are the inputs for this three instances of GCN by Eq 2.2. Each GCN instance generates a batch of vector

$Z^i \in R^{N \times k}$, where k represents the dimension of the final hidden layer output by GCN, and Z^i signifies the output of the i -th instance. While the value of k can be different for each GCN, for simplicity, let's assume that all k values are set to be the same. Subsequently, we merge the outputs of these three GCNs and feed them into a classification network, thereby enabling joint training of all GCNs and the classification network through backpropagation, where the number of predicted categories are represented as c . From a deep learning perspective, the classification network can be intuitively represented as a fully connected layer. We employ a fully connected layer $FCN_{fc} : R^{N \times 3k} \rightarrow R^{N \times c}$ with a trainable parameter matrix $W^{fc} \in R^{3k \times c}$, which can be expressed as follows:

$$FCN_{fc} = \text{Softmax}(\text{Concat}(\text{GCN}(X, A), \text{GCN}(X^{dm}(n_1), A^{dm}(n_1)), \text{GCN}(X^{dm}(n_2), A^{dm}(n_2))))W^{fc} \quad (3.8)$$

The classification network has the capability to select features from each of the GCN instances, thus serving the purpose of fusing different-scale features. To enhance the generalization ability of the classification network, we employ self-attention mechanisms [42,43] to automatically learn the weights associated with the node features obtained from the three GCN instances. Hence, the classification network can adaptively adjust the weights of node features from different scales, with a focus on the most salient scale, thereby allowing for better fitting of the labels for input nodes.

The fundamental idea of self-attention in the classification network can be illustrated using Figure 2, where Z^1 , Z^2 and Z^3 represent the integrated node features at three different scales achieved through self-attention. The classification network simultaneously employs the outputs of the three GCN instances and concatenates the results, thus obtaining fused node features from different scales; then, Z^1 , Z^2 and Z^3 are input into a fully connected layer, and go to learn the attention weights (α, β, γ) respectively. Specifically, attention weights can be calculated according to the following:

$$(\alpha, \beta, \gamma) = \text{Softmax}(\text{Concat}(Z^1, Z^2, Z^3) \cdot W^{att}) \quad (3.9)$$

Here, W^{att} represents the attention weight matrix. Once the attention weights are obtained, the self-attention module multiplies the node features at each scale by their respective weights. These weighted features are concatenated and fed into a fully connected network represented by Eq 3.8 to obtain the output of the classification network, where different-scale node features are fused for each node's category-probability vector of Z .

$$Z = \text{Softmax}(\text{Concat}(\alpha Z^1, \beta Z^2, \gamma Z^3) \cdot W^{fc}) \quad (3.10)$$

Finally, we employ the cross-entropy loss function for model training, which is defined as follows:

$$\text{Loss} = - \sum_{i \in N} \sum_{j=1}^c Y_{ij} \ln Z_{ij} \quad (3.11)$$

Here, Y represents the label set corresponding to the labeled sample data in the current training dataset, Y_{ij} denotes the label at the corresponding position of the node in the training dataset, and Z_{ij} represents the output value for the corresponding node in the current network output. For clarity, the detailed procedure of the proposed MFF-GCN is presented in Algorithm 1.

Algorithm 1: MFF-GCN

Input: $X \in R^{N \times d}$, label information Y , Number of neighbors n_1, n_2 , and Number of trainings T

- 1 initialization: Three GCN parameters W_1^l, W_2^l, W_3^l , Metric learning parameters W^d , FCN_{fc} parameters W^{fc} , self-attention parameters W^{att} .
- 2 constructed graph: $G = (X, A) \leftarrow X$ by (3.1) and (3.2).
- 3 Diffusion maps: $X^{dm}(n_1), X^{dm}(n_2) \leftarrow \{X, n_1, n_2\}$ by (3.1), (3.3), (3.4) and (3.5).
- 4 while epoch $< T$ do
- 5 $A^{dm}(n_1), A^{dm}(n_2) \leftarrow \{X^{dm}(n_1), X^{dm}(n_2), W^d\}$ by (3.1), (3.6) and (3.7)
- 6 $Z^1, Z^2, Z^3 \leftarrow \{X, A, W_1^l, X^{dm}(n_1), A^{dm}(n_1), W_2^l, X^{dm}(n_2), A^{dm}(n_2), W_3^l\}$ by (2.2)
- 7 $\alpha, \beta, \gamma \leftarrow \{Z^1, Z^2, Z^3, W^{att}\}$ by (3.9)
- 8 $Z \leftarrow \{Z^1, Z^2, Z^3, \alpha, \beta, \gamma, W^{fc}\}$ by (3.10)
- 9 Loss $\leftarrow \{Y, Z\}$ by (3.11)
- 10 end while

Output: Z

4. Experiments

4.1. Data pre-processing

In our experiments, we obtained 1.5T T1-weighted MRI images from the publicly available ADNI dataset (adni.loni.usc.edu). We sequentially processed the MRI images using SPM8 and the DPABI toolbox, which involved the following steps: removing non-brain tissue, motion and time correction, spatial normalization, filtering, and smoothing. Subsequently, we performed brain tissue segmentation on the MRI images, thus resulting in gray matter, white matter, and cerebrospinal fluid. Then these segmented regions were registered to the automated anatomical labeling (AAL) template to obtain the corresponding regions of interest (ROIs) within the brain.

Since we used the AAL human brain atlas template for partitioning the brains of the subjects, this atlas template divides the brain into 116 regions. Therefore, the overall representation obtained after partitioning the brains of the subjects is a 116-dimensional vector.

Table 1. The number of samples in the ADNI dataset.

Dataset	Samples	Dataset	Samples
AD-CN	338: 532	CN-MCI	532: 422
AD-MCI	338: 422	AD-MCI-CN	338: 422: 532

In this study, we obtained a total of 1292 samples, with 338 samples from AD, 422 samples from MCI, and 532 samples from the CN, which consists of individuals without any cognitive impairments. We evaluated the classification performance of the proposed method on four datasets, including AD vs CN (discriminating between AD and CN), AD vs MCI (discriminating between AD and MCI), CN vs MCI (discriminating between CN and MCI), and AD vs MCI vs CN (discriminating among AD, MCI, and CN). Table 1 displays the distribution of data samples for each dataset.

4.2. Experimental setup

We conducted our experiments on a server equipped with an NVIDIA GeForce 3060ti (8 GB) GPU, using the PyTorch framework and PyTorch Geometric library. We performed multiple random experiments on each of the four datasets, and the reported results are based on their average performance. We employed the widely used metric of classification accuracy to evaluate all methods.

4.3. Setup of hyperparameters

We adjusted the hyperparameters for each method by referring to the relevant literature to obtain their best results. For our method, we set the maximum number of epochs to 500, the learning rate to 0.001, used the Adam optimizer, and applied the cross-entropy loss function. We utilized a two-layer GCN in all of our GCNs, with l set to 2, thus indicating two layers of GCN. The dataset splits used in all four experiments were in a ratio of 7:1:2 (training set: validation set: test set).

4.4. Comparative experiments

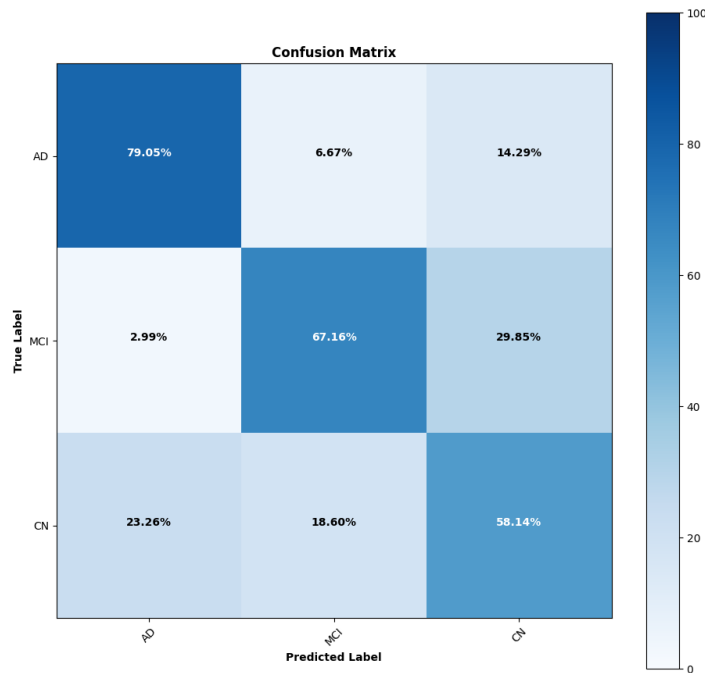


Figure 3. Confusion matrices.

To validate the effectiveness of our proposed method, we conducted a comprehensive comparison analysis against several baseline methods. In the experiments, we used the same datasets for training and the same test sets for comparison. Additionally, we employed standard metrics widely used in academia and industry, including classification accuracy, to provide an objective performance assessment of our method across the four datasets. The experimental results, organized according to the classification tasks on different datasets, are presented in Table 2. The experiments we compared against are listed in the table, including traditional machine learning methods such as SVM [44], image classification methods like ResNet [45], other node classification methods in GNNs such as

GCN [29] and GAT [46], and graph classification methods in GNNs such as Graph U-net [47], SAG Pooling [48], and Dir-GNN [49].

Table 2. The Classifications Performance on AD vs CN, AD vs MCI, CN vs MCI and AD VS MCI VS CN Datasets.

Method	AD-CN	AD-MCI	CN-MCI	AD-MCI-CN
SVM	0.614	0.562	0.554	0.407
ResNet	0.615	0.653	0.742	0.640
GCN	0.789	0.594	0.692	0.526
GAT	0.829	0.636	0.699	0.551
Graph U-net	0.821	0.584	0.709	0.644
SAGPooling	0.838	0.654	0.736	0.652
DirGNN	0.852	0.594	0.737	0.640
DM-GCN	0.866	0.664	0.720	0.555
Our	0.887	0.705	0.825	0.671

Table 2 presents the classification performance of all methods on the four datasets, with the best results for each metric highlighted in bold. From the table, it's evident that our method achieved the best average classification accuracy across all four datasets. In the AD – CN classification task, our method achieved an average classification accuracy of 88.7%, which is at least 2% higher than other methods. In the challenging AD-MCI-CN three-class classification task, our method also performed well, with an average classification accuracy of 66.7%. Figure 3 displays the confusion matrices, thereby providing a visual representation of how well the predictions match the actual categories. In the plotted confusion matrices, you can see darker diagonal cells, thus indicating higher accuracy. Model misclassifications are represented by off-diagonal elements with shading.

As shown in Figure 4, this illustrates the training loss curves of our proposed method when compared to other GNNs approaches. The graph reveals that our method achieves lower training losses compared to others. Additionally, our method demonstrates a faster convergence, thus indicating an improved training efficiency in contrast to the other approaches.

Our method demonstrates a strong performance across multiple classification tasks due to the multi-scale node feature fusion based on diffusion maps. By fusing node features obtained from the diffusion maps at different scales, we effectively refine important information while reducing the impact of redundant information on classification results. This approach leads to excellent classification outcomes.

4.5. Ablation experiments

To investigate the impact of the diffusion maps module, the metric learning module, and the multiscale feature fusion module on the final model's classification results, we conducted experiments on the test dataset by individually removing each of these three components. This helps us understand the contribution of each module to the overall improvement in method performance.

As shown in Table 3, we can observe that when the multi-scale feature fusion module and metric learning module are not used, the AD – CN classification performance of the DM + GCN model

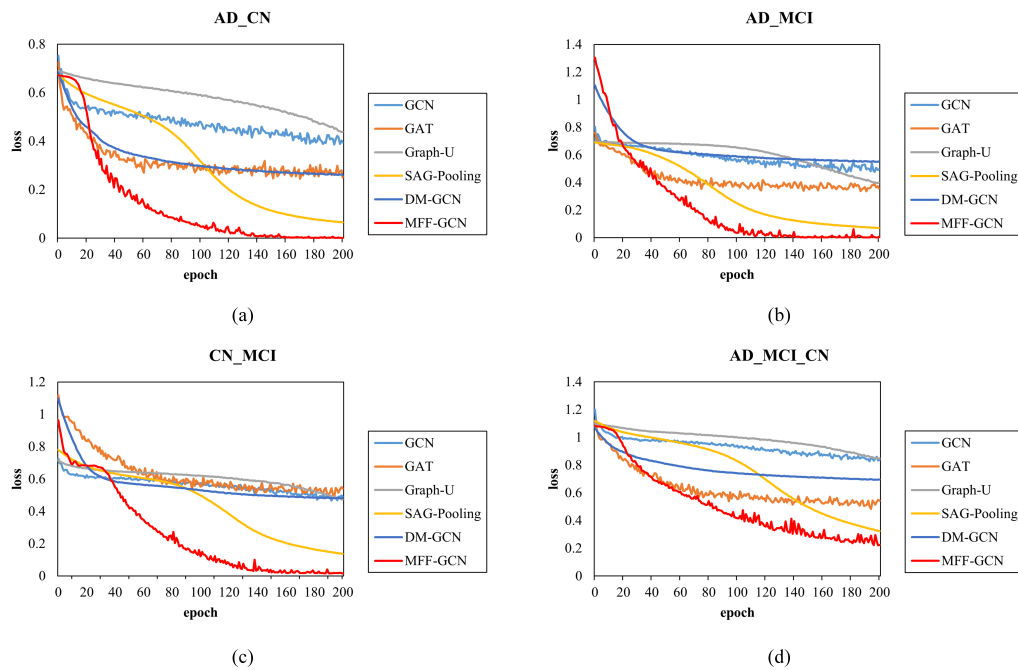


Figure 4. The training loss curves of the proposed GNNs method for: (a) AD_CN dataset. (b) AD_MCI dataset. (c) CN_MCI dataset. (d) AD_CN_MCI dataset.

Table 3. Table of ablation experiments.

Module	AD-CN	AD-MCI	CN-MCI	AD-MCI-CN
GCN	0.789	0.594	0.692	0.526
GCN + DM	0.866	0.664	0.720	0.555
GCN + DM + ML	0.874	0.692	0.801	0.653
GCN + DM + FF	0.865	0.651	0.777	0.565
GCN + ML + DM + FF	0.887	0.705	0.825	0.671

decreases by 2%. Similarly, when the metric learning module is not used, the AD CN classification performance of the DM + FF + GCN model still decreases by 2%. This indicates that the effect of multi-scale feature fusion is not significant without the metric learning module to adjust the input graph structure. When the multi-scale feature fusion module is not used, the AD – CN classification performance of the DM + ML + GCN model decreases by 1%, thus highlighting the role of this module in feature fusion. If we do not obtain diffusion node features through the diffusion mapping module, there is no need to use the metric learning module to adjust the input graph structure. The multiscale feature fusion module relies on the features generated by the diffusion mapping module and cannot be used in isolation. The results of ablation experiments on the other three datasets are similar.

In summary, the three modules introduced in our approach have played a pivotal role in enhancing the model's ultimate classification performance. The metric learning module has heightened the multi-scale feature fusion module's capacity to perceive multi-scale features, while the diffusion map module has aggregated feature information from nodes with similar characteristics in the feature space. By incorporating these modules, our model exhibits an improved adaptability to various classification

tasks, thus showcasing a heightened generalization performance and robustness.

5. Conclusions

In this research endeavor, we have introduced a graph convolutional classification network that leverages the principles of diffusion map for the fusion of multi-scale node characteristics. This augmentation is aimed at enhancing the classification of Alzheimer's disease based on MRIs. The methodology commences by extracting cerebral features from MRI scans, segmented according to brain regions. Each patient's MRI characteristics are construed as individual nodes, culminating in the formation of a network representing Alzheimer's disease patients. By applying diffusion maps to patient node characteristics, combined with the integration of diverse-scale node feature fusion techniques, we ultimately achieved the classification and diagnosis of Alzheimer's disease. When juxtaposed with state-of-the-art methodologies, our proposed network architecture exhibited a commendable classification performance across multiple datasets. Throughout this study, we solely utilized MRI as the medical data source. This limitation might affect the model's expressive capability, thereby influencing its performance. In future research, we aim to explore our approach's performance using multimodal medical data, including, but not limited to, fMRI and DTI, while considering the addition of non-medical information for study purposes. In terms of the methodology, our current model's diffusion in the feature space is based on unsupervised manifold learning, thereby relying on fine-tuning hyperparameters that lack adaptability. In future work, we aim to incorporate supervised manifold learning into information diffusion within the feature space.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant NO. 62306106, Natural Science Foundation of Hubei Province under Grant NO. 2023AFB377, Doctoral Research Start-up Foundation of Hubei University of Technology under Grant NO. XJ2022007301, the Doctoral Starting up Foundation in Hubei University of Techonology under Grant NO. XJ2022006402.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. G. McKhann, D. Drachman, M. Folstein, R. Katzman, D. Price, E. M. Stadlan, Clinical diagnosis of Alzheimer's disease: Report of the NINCDS-ADRDA Work Group* under the auspices of Department of Health and Human Services Task Force on Alzheimer's disease, *Neurology*, **34** (1984), 939–939. <https://doi.org/10.1212/WNL.34.7.939>

2. L. F. Jia, M. N. Quan, Y. Fu, T. Zhao, Y. Li, C. B. Wei, et al., Dementia in China: epidemiology, clinical management, and research advances, *Lancet Neurol.*, **19** (2020), 81–92. [https://doi.org/10.1016/S1474-4422\(19\)30290-X](https://doi.org/10.1016/S1474-4422(19)30290-X)
3. Risk reduction of cognitive decline and dementia: WHO guidelines, *World Health Organization*, 2019. Available from: <https://www.who.int/publications-detail-redirect/9789241550543>.
4. M. Calabrò, C. Rinaldi, G. Santoro, C. Crisafulli, The biological pathways of Alzheimer disease: A review, *AIMS Neurosci.*, **8** (2021), 86–86. <https://doi.org/10.3934/Neuroscience.2021005>
5. W. Jagust, Vulnerable neural systems and the borderland of brain aging and neurodegeneration, *Neuron*, **77** (2013), 219–234. <http://dx.doi.org/10.1016/j.neuron.2013.01.002>
6. N. Habib, C. McCabe, S. Medina, M. Varshavsky, D. Kitsberg, R. Dvir-Szternfeld, et al., Disease-associated astrocytes in Alzheimer’s disease and aging, *Nat. Neurosci.*, **23** (2020), 701–706. <https://doi.org/10.1038/s41593-020-0624-8>
7. Alzheimer’s Association, 2019 Alzheimer’s disease facts and figures, *Alzheimer Dementia*, **15** (2019), 321–387. <https://doi.org/10.1016/j.jalz.2019.01.010>
8. J. H. Wen, E. Thibeau-Sutre, M. Diaz-Melo, J. Samper-González, A. Routier, S. Bottani, et al., Convolutional neural networks for classification of Alzheimer’s disease: Overview and reproducible evaluation, *Med. Image Anal.*, **63** (2020), 101694. <https://doi.org/10.1016/j.media.2020.101694>
9. M. H. Liu, F. Li, H. Yan, K. D. Wang, Y. X. Ma, L. Shen, et al., A multi-model deep convolutional neural network for automatic hippocampus segmentation and classification in Alzheimer’s disease, *Neuroimage*, **208** (2020), 116459. <https://doi.org/10.1016/j.neuroimage.2019.116459>
10. T. Abuhmed, S. El-Sappagh, J. M. Alonso, Robust hybrid deep learning models for Alzheimer’s progression detection, *Knowl. Based Syst.*, **213** (2021), 106688. <https://doi.org/10.1016/j.knosys.2020.106688>
11. A. M. Alvi, S. Siuly, H. Wang, K. Wang, F. Whittaker, A deep learning based framework for diagnosis of mild cognitive impairment, *Knowl. Based Syst.*, **248** (2022), 108815. <https://doi.org/10.1016/j.knosys.2022.108815>
12. X. M. Chen, T. Wang, H. R. Lai, X. L. Zhang, Q. J. Feng, M. Y. Huang, Structure-constrained combination-based nonlinear association analysis between incomplete multimodal imaging and genetic data for biomarker detection of neurodegenerative diseases, *Med. Image Anal.*, **78** (2022), 102419. <https://doi.org/10.1016/j.media.2022.102419>
13. G. B. Frisoni, N. C. Fox, C. R. Jack Jr, P. Scheltens, P. M. Thompson, The clinical use of structural MRI in Alzheimer disease, *Nat. Rev. Neurol.*, **6** (2010), 67–77. <https://doi.org/10.1038/nrneurol.2009.215>
14. P. Cao, X. L. Liu, J. Z. Yang, D. Z. Zhao, M. Huang, J. Zhang, et al., Nonlinearity-aware based dimensionality reduction and over-sampling for AD/MCI classification from MRI measures, *Comput. Biol. Med.*, **91** (2017), 21–37. <https://doi.org/10.1016/j.combiomed.2017.10.002>
15. K. Bäckström, M. Nazari, I.Y. Gu, A.S. Jakola, An efficient 3D deep convolutional network for Alzheimer’s disease diagnosis using MR images, in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, (2018), 149–153. <https://doi.org/10.1109/ISBI.2018.8363543>

16. C. F. Lian, M. X. Liu, L. Wang, D. G. Shen, End-to-end dementia status prediction from brain mri using multi-task weakly-supervised attention network, in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part IV 22*, (2019), 158–167. https://doi.org/10.1007/978-3-030-32251-9_18
17. K. Mortensen, T. L. Hughes, Comparing Amazon’s Mechanical Turk platform to conventional data collection methods in the health and medical research literature, *J. Gene. Intern. Med.*, **33** (2018), 533–538. <https://doi.org/10.1007/s11606-017-4246-0>
18. S. Parisot, S. I. Ktena, E. Ferrante, M. Lee, R. Guerrero, B. Glocker, et al., Disease prediction using graph convolutional networks: application to autism spectrum disorder and Alzheimer’s disease, *Med. Image Anal.*, **48** (2018), 117–130. <https://doi.org/10.1016/j.media.2018.06.001>
19. L. Peng, R. Y. Hu, F. Kong, J. Z. Gan, Y. J. Mo, X. S. Shi, et al., Reverse graph learning for graph neural network, *IEEE Trans. Neural Networks Learn. Syst.*, (2022). <https://doi.org/10.1109/TNNLS.2022.3161030>
20. Q. Ni, J. C. Ji, B. Halkon, K. Feng, A. K. Nandi, Physics-Informed Residual Network (PIResNet) for rolling element bearing fault diagnostics, *Mechan. Syst. Signal Process.*, **200** (2023), 110544. <https://doi.org/10.1016/j.ymsp.2023.110544>
21. Y. D. Xu, K. Feng, X. A. Yan, R. Q. Yan, Q. Ni, B. B. Sun, et al., CFCNN: A novel convolutional fusion framework for collaborative fault identification of rotating machinery, *Inf. Fusion*, **95** (2023), 1–16. <https://doi.org/10.1016/j.inffus.2023.02.012>
22. Z. Yang, K. Li, H. T. Gan, Z. W. Huang, M. Shi, HD-GCN: A hybrid diffusion graph convolutional network, preprint, arXiv:2303.17966.
23. B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in *Proceedings of the 20nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2014), 701–710. <https://doi.org/10.1145/2623330.2623732>
24. A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2016), 855–864. <https://doi.org/10.1145/2939672.2939754>
25. S. Abu-El-Haija, B. Perozzi, R. Al-Rfou, A. A. Alemi, Watch your step: Learning node embeddings via graph attention, in *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, 2018.
26. S. Abu-El-Haija, A. Kapoor, B. Perozzi, J. Lee, N-gcn: Multi-scale graph convolution for semi-supervised node classification, *Uncertainty Artif. Intell.*, **2020** (2020), 841–851.
27. J. L. Zhu, M. W. Jia, Y. Zhang, W. H. Zhou, H. Y. Deng, Y. Liu, Domain adaptation graph convolution network for quality inferring of batch processes, *Chemom. Intell. Lab. Syst.*, **2023** (2023), 105028. <https://doi.org/10.1016/j.chemolab.2023.105028>
28. M. W. Jia, D. Y. Xu, T. Yang, Y. Liu, Y. Yao, Graph convolutional network soft sensor for process quality prediction, *J. Process Control*, **123** (2023), 12–25. <https://doi.org/10.1016/j.jprocont.2023.01.010>

29. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, preprint, arXiv:1609.02907.
30. D. Liben-Nowell, J. Kleinberg, The link prediction problem for social networks, in *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, **2003** (2003), 556–559. <https://doi.org/10.1145/956863.956972>
31. A. Fout, J. Byrd, B. Shariat, A. Ben-Hur, Protein interface prediction using graph convolutional networks, in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, (2017), 1–10.
32. T. Hamaguchi, H. Oiwa, M. Shimbo, Y. Matsumoto, Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach, preprint, arXiv:1706.05674.
33. E. Xing, M. Jordan, S. J. Russell, A. Ng, Distance Metric Learning with Application to Clustering with Side-Information, in *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, (2002), 521–528.
34. K. Q. Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin nearest neighbor classification, in *Advances in Neural Information Processing Systems 18 (NIPS 2005)*, (2005), 1473–1480.
35. R. Y. Li, S. Wang, F. Y. Zhu, J. Z. Huang, Adaptive graph convolutional neural networks, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **32** (2018). <https://doi.org/10.1609/aaai.v32i1.11691>
36. S. G. Lv, G. Wen, S. Y. Liu, L. S. Wei, M. Li, Robust graph structure learning with the alignment of features and adjacency matrix, preprint, arXiv:2307.02126.
37. S. Parisot, S. I. Ktena, E. Ferrante, M. Lee, R. Guerrero, B. Glocker, et al., Disease prediction using graph convolutional networks: application to autism spectrum disorder and Alzheimer’s disease, *Med. Image Anal.*, **48** (2018), 117–130. <https://doi.org/10.1016/j.media.2018.06.001>
38. A. Kazi, S. Shekarforoush, K. Kortuem, S. Albarqouni, N. Navab, Self-attention equipped graph convolutions for disease prediction, in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, (2019), 1896–1899. <https://doi.org/10.1109/ISBI.2019.8759274>
39. A. Kazi, S. Shekarforoush, S. Arvind Krishna, H. Burwinkel, G. Vivar, B. Wiestler, et al., Graph convolution based attention model for personalized disease prediction, in *Medical Image Computing and Computer Assisted Intervention*, (2019), 122–130. https://doi.org/10.1007/978-3-030-32251-9_14
40. A. Kazi, S. Shekarforoush, S. Arvind Krishna, H. Burwinkel, G. Vivar, K. Kortüm, et al., InceptionGCN: receptive field aware graph convolutional network for disease prediction, in *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019*, (2019), 73–85. https://doi.org/10.1007/978-3-030-20351-1_6
41. G. Vivar, A. Zwergal, N. Navab, S. A. Ahmadi, Multi-modal disease classification in incomplete datasets using geometric matrix completion, in *Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities*, (2018), 24–31. https://doi.org/10.1007/978-3-030-00689-1_3
42. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

43. V. Mnih, N. Heess, A. Graves, Recurrent models of visual attention, in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 2014.
44. C. Cortes, V. Vapnik, Support-vector networks, *Mach. Learn.*, **20** (1995), 273–297. <https://doi.org/10.1007/BF00994018>
45. K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 770–778.
46. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, preprint, arXiv:1710.10903 .
47. H. Y. Gao, S. W. Ji, Graph u-nets, in *international Conference on Machine Learning*, (2019), 2083–2092.
48. J. Lee, I. Lee, J. Kang, Self-attention graph pooling, in *International Conference on Machine Learning*, (2019), 3734–3743.
49. E. Rossi, B. Charpentier, F. Di Giovanni, F. Frasca, S. Günnemann, M. Bronstein, Edge directionality improves learning on heterophilic graphs, preprint, arXiv:2305.10498



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)