Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 8342104, 28 pages
https://doi.org/10.1155/2023/8342104

Review Article

Deep Learning with Graph Convolutional Networks: An Overview and Latest Applications in Computational Intelligence

Uzair Aslam Bhatti,[1,2] **Hao Tang**,[1] **Guilu Wu**,[1] **Shah Marjan**,[3] **and Aamir Hussain**[4]

[1] *School of Information and Communication Engineering, Hainan University, Haikou 570100, China*
[2] *State Key Laboratory of Marine Resource Utilization in the South China Sea, Hainan University, Haikou, China*
[3] *Department of Software Engineering, Balochistan University of Information Technology, Engineering, and Management Sciences (BUITEMS), Quetta, Pakistan*
[4] *Department of Computer Science, Muhammad Nawaz Shareef University of Agriculture Multan, Multan, Pakistan*

Correspondence should be addressed to Hao Tang; melineth@hainanu.edu.cn

Received 22 August 2022; Revised 8 December 2022; Accepted 9 February 2023; Published 28 February 2023

Academic Editor: Subrata Kumar Sarker

Convolutional neural networks (CNNs) have received widespread attention due to their powerful modeling capabilities and have been successfully applied in natural language processing, image recognition, and other fields. On the other hand, traditional CNN can only deal with Euclidean spatial data. In contrast, many real-life scenarios, such as transportation networks, social networks, reference networks, and so on, exist in graph data. The creation of graph convolution operators and graph pooling is at the heart of migrating CNN to graph data analysis and processing. With the advancement of the Internet and technology, graph convolution network (GCN), as an innovative technology in artificial intelligence (AI), has received more and more attention. GCN has been widely used in different fields such as image processing, intelligent recommender system, knowledge-based graph, and other areas due to their excellent characteristics in processing non-European spatial data. At the same time, communication networks have also embraced AI technology in recent years, and AI serves as the brain of the future network and realizes the comprehensive intelligence of the future grid. Many complex communication network problems can be abstracted as graph-based optimization problems and solved by GCN, thus overcoming the limitations of traditional methods. This survey briefly describes the definition of graph-based machine learning, introduces different types of graph networks, summarizes the application of GCN in various research fields, analyzes the research status, and gives the future research direction.

1. Introduction

AI has drawn the attention of the whole industry in recent years as a frontier field of scientific research and has progressively become a new engine for social and economic development [1]. NLP, computer vision, intelligent robots, data mining, cognition and reasoning, and other disciplines of society have widely practiced and implemented AI. Today's network data traffic is increasing dramatically due to the rapid expansion of smart devices such as smartphones, smart automobiles, and smart homes. Simultaneously, using technologies such as edge computing, virtualization, and network slicing broadens network services, improves user experience, and creates a more complicated network environment. The efficient management of a large number of intelligent devices and the optimization of resource allocation in large-scale and complex network environments has emerged as a critical challenge for future network growth. AI, as the future network's brain, is employed for network optimization and decision-making. Simultaneously, increasing the computational capability of network nodes adds bones and muscles to the network, allowing intelligent network calculations to be performed [2]. Besides that, the image processing field is also improving day by day due to different high-dimensional and complex images generated from different data sources. Deep learning applications have attracted great attention in the field of image processing due to their ultrahigh prediction accuracy in recognition applications, which is bound to improve the performance of existing image processing systems and open up new application fields [3–5].

The coordinated development of algorithms and computing power will enable future applications to enter a new era of intelligence. Graph data is a typical non-Euclidean spatial data with complex correlations and interobject dependencies [6]. The traditional graph theory method is difficult to adapt to the complex graph problems in the future network. Therefore, finding algorithms to solve complex graph data to guide the resource allocation, management, and scheduling of communication networks has become an important scientific problem in future networks. As an emerging technology in the field of AI in recent years, GNN has opened up a new space for processing complex graph structure data. With the help of artificial intelligence technologies such as deep learning and reinforcement learning, GCN can quickly mine topological information and complex features in graph structures and have solved many major problems in the fields of computer vision, recommendation systems, and knowledge graphs [7, 8]. Therefore, the combination of GCN and the latest advancement is an important way to solve real-world issues efficiently and effectively.

1.1. Significance of this survey.

CNN has grown dramatically in recent years, attracting worldwide attention due to its remarkable modeling skills. In comparison to older methods, the introduction of CNN has resulted in significant advances in the disciplines of image processing and natural language processing, such as machine translation, image recognition, and speech recognition, among others [9]. Traditional convolutional neural networks, on the other hand, can only deal with data in Euclidean space (such as images, text, and speech), and the data in these domains is translation invariant. Because of translation invariance, we can build a globally shared convolution kernel in the input data space, allowing us to define a convolutional neural network. Using picture data as an example, an image can be represented as a set of regularly distributed pixels in Euclidean space, and translation invariance means that local structures of the same size can be produced with any pixel as the center [10]. Based on this, the CNN develops meaningful hidden layer representations for pictures by learning convolution kernels shared at each pixel and then models local connections. Figure 1 depicts the distinction between non-Euclidean and Euclidean space.

Although traditional CNN bring improvements in text and image domains, they can only handle Euclidean space data. At the same time, non-Euclidean spatial data: graph data, has gradually attracted attention due to its ubiquity. Graph data can naturally express data structures in real life, such as transportation networks, World Wide Web, and social networks. Different from image and text data, the local structure of each node in graph data are different, which makes the translation invariance no longer satisfied [11]. The lack of translation invariance poses a challenge to define CNN on graph data. In recent years, due to the ubiquity of graph data, researchers have begun to focus on how to construct deep-learning models on graphs. With the help of the CNN's ability to model local structures and the ubiquitous node dependencies on the graph, the GCN has become the most active and important algorithm. Recently, some articles have emerged to explore and summarize deep learning on graphs, but for the most important branch, GCN, the in-depth discussion and summary of its modeling methods and applications are still an area that needs to be focused on. In this regard, in this article, we deeply organize and summarize the development history and future trends of GCN with applications developed recently in all fields of science. The challenges faced in the construction of GCN mainly come from the following aspects:

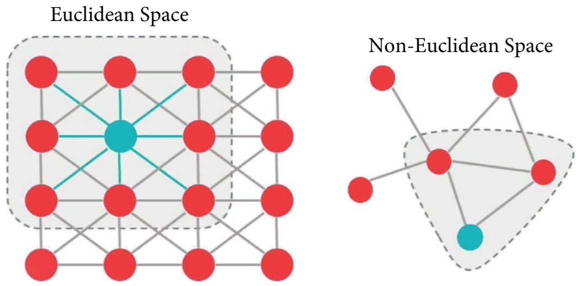


FIGURE 1: Highlighting the difference between euclidean space and non-euclidean space.

(i) Graph data are non-European spatial data:

Graph data, as non-Euclidean spatial data, does not satisfy translation invariance, which means that each node has different local structures. Traditional convolutional neural networks' basic operators, convolution and pooling, rely on data translation invariance. At the moment, defining convolution and pooling operators on graph data are a difficult task.

(ii) Different characteristics of graph data:

Graph data can be used to represent a wide range of real-world applications, such as social networks, citation networks, and political relationships networks, all of which have unique characteristics that can be represented by graph data. Positive and negative tendencies are linked to signs, symbols, and other indicators. GCNs are more difficult to design because they have to model a wider range of graph characteristics.

(iii) Graph data with large scale:

Large-scale graphs, such as user commodity networks and user networks in social networks, can be used in practical applications with millions or even tens of millions of nodes in them. Additionally, the challenge of building a large-scale graph convolutional neural network with acceptable time and space constraints is a major one.

The current survey's primary focus is GCN, including its variants and the most recent GCN trends. We specifically cover all of the most recent works that use the GCN in various fields of science. We concentrate on works published between the years 2000 and 2022. We used the PRISMA (preferred reporting items for systematic reviews and meta-analyses) framework guidelines to select GCN-related publications. Papers were found through a variety of sources of publishers, including Springer, IEEE, MDPI, HINDWAI, WILEY, Elsevier, and ACM library. Articles were searched with different titles as follows: "graph convolutional networks," "graph networks," "GCN," "Graph attention network," "Attention-based Graph," "GAT," "GATnet," and "Graph Query." All the articles were searched only in the English language in the selected time period. The main contributions of this survey are three-fold as follows:

(1) We provide a thorough analysis of GCN deep learning techniques, including variants and advancements in GCN, applications, and current trends in various fields of study, performance measures, and so on

(2) A hierarchical and structural review of recent improvements in deep learning-based GCN techniques is offered, and the benefits and limitations of each component for an effective GCN solution are examined

(3) In order to provide intelligent, advice to the general public, we discuss the obstacles and unresolved concerns, as well as new trends and future directions

Figure 2 gives the complete structure of our survey. This paper firstly introduces the basic model of GNN and several important graph neural networks; secondly, it introduces the specific application methods of GNN in various fields of research such as NLP, computer vision etc; in the conclusion part, it discusses the current research status and gives the future research direction.

2. Graph Neural Network (GNN) and Its Variant

GNN was first proposed by Gori et al. [12] and Scarselli et al. [13] elaborated on this model in more detail. GNN proposed by Gori et al. [12] draws on the research results in the field of neural networks, which can directly process graph structure data, and its core is the local transfer function and the local output function. The local transfer function generates the state vector of the node, which contains the neighborhood information of the node. The transfer function is shared among all nodes and updates the node's state vector h_1 according to the input neighborhood, and its expression is as follows:

$$\mathbf{h}_v = f\left(\mathbf{x}_v, \mathbf{x}_{vu}^{\mathrm{e}}, \mathbf{h}_u, \mathbf{x}_u\right). \tag{1}$$

In the formula, $\mathbf{x}_v$ is the feature of the node $\mathbf{x}_{vu}^{\mathrm{e}}$ is the feature connecting the node v and its neighbor node u, $\mathbf{x}_u$ is the feature of the neighbor node of the node v. The local output function generates a new representation of the node, and its expression is as follows:

$$\mathbf{o}_v = g\left(\mathbf{h}_v, \mathbf{x}_v\right). \tag{2}$$

The stacking form of the local transfer function and the local output function applied to all nodes constitutes a GNN structure model that will eventually reach a stable state through iteration. The early graph neural network has great limitations, its efficiency is low, the computational cost is high, and the node characteristics. It is difficult to affect the state after multiple updates. In recent years, in order to process graph structure data more efficiently, new graph neural networks and application studies have been proposed one after another.

2.1. Graph Convolutional Networks (GCNs). GCN introduces convolution operation into graph structure and is one of the most important GNNs at present. According to the different feature extraction methods, it can be divided into GCN based on spectral-domain and graph convolution network based on the spatial domain. The graph convolution network are derived from graph signal processing, and a filter is introduced to define graph convolution, which can be understood as removing noise through a filter to obtain the classification result of the input signal.

Based on the spectral graph theory, Bruna et al. first proposed the convolution layer function to define the spectral domain GCN [14]. Kipf and Welling [15] first proposed the concept of semisupervised GCN based on the spectral domain (structure shown in Figure 3). The spectral domain graph convolution is defined as the product of the signal and the filter function, and its expression is as follows:

$$g_{\boldsymbol{\theta}} * x = \mathbf{U} g_{\boldsymbol{\theta}} \mathbf{U}^T x. \tag{3}$$

In the formula, g_θ is the filter function, x is the signal of the graph at the node, and U is the eigenvector of the normalized Laplacian matrix of the graph. g_θ can be understood as the eigenvalue function of the graph Laplacian matrix, namely, $g_\theta\ (\Lambda)$, where Λ is the diagonal matrix composed of the eigenvalues of the graph Laplacian matrix, and θ is the function parameter. In order to reduce the computational complexity, $g_\theta\ (\Lambda)$ can be approximated, and its expression is as follows:

$$\begin{aligned} g_{\boldsymbol{\theta}'}(\boldsymbol{\Lambda}) &\approx \sum_{k=0}^{k} \boldsymbol{\theta}_k' T_k(\overline{\mathbf{L}}), \\ \overline{\mathbf{L}} &= \frac{2}{\lambda_{\max}} \mathbf{L} - \mathbf{I}_N, \\ \mathbf{L} &= \mathbf{I}_N - \mathbf{D}^{-(1/2)} \mathbf{A} \mathbf{D}^{-(1/2)}. \end{aligned} \tag{4}$$

In the formula, T_k is the k-order Chebyshev polynomial, $\theta^{'}$ is the Chebyshev coefficient vector, L is the graph Laplacian matrix, $\lambda_{\max}$ is the largest eigenvalue of L, I_N is the identity matrix, D is the opposite angle matrix, and A is an adjacency matrix. When limiting $k = 1$, the convolutional layer can be simplified to as follows:

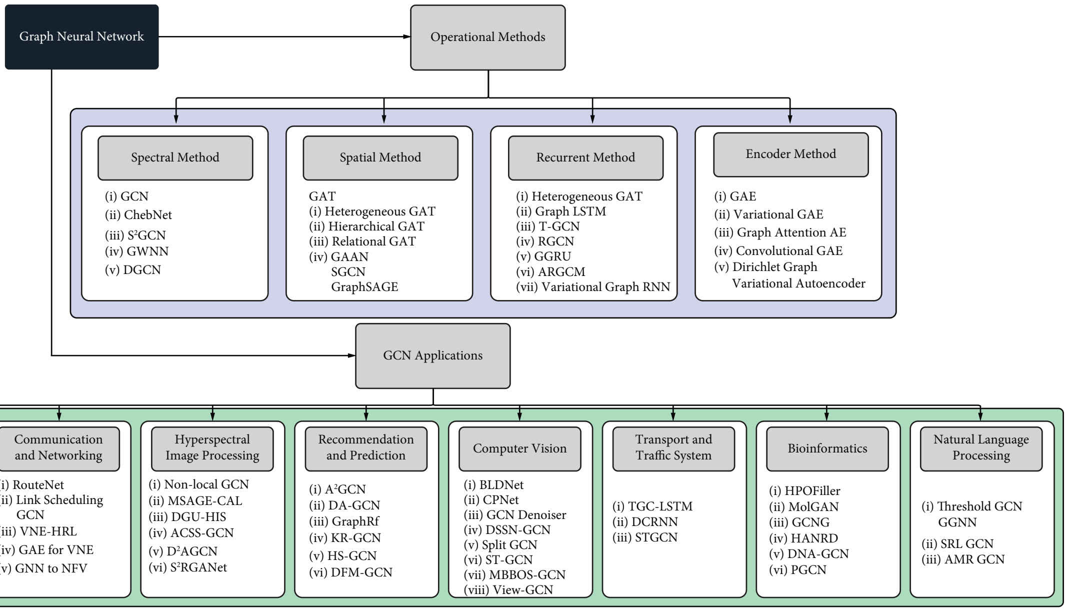


Figure 2: Hierarchically-structured taxonomy of this survey.

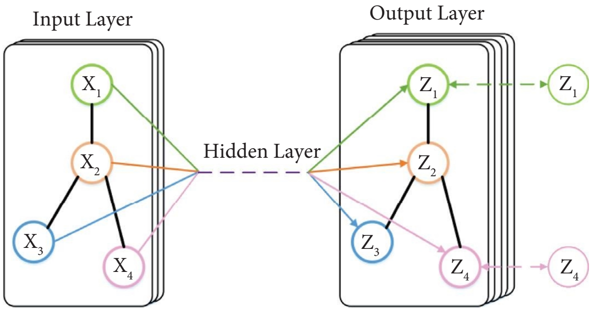


Figure 3: Basic structural form of GCN.

$$
\begin{aligned}
g_{\boldsymbol{\theta}}^{*}\mathbf{x} &\approx \boldsymbol{\theta}\left(\mathbf{I}_N + \mathbf{D}^{-(1/2)} A \mathbf{D}^{-(1/2)}\right)\mathbf{x},\\
\overline{A} &= A + I_N,\\
\overline{D}_{ii} &= \sum_j \overline{A}_{ij}.
\end{aligned}
\tag{5}
$$

Then, the convolutional layer formula of the graph convolutional network is as follows:

$$
\mathbf{h}^{(l)} = \sigma\left(\overline{\mathbf{D}}^{-(1/2)}\overline{\mathbf{A}}\overline{\mathbf{D}}^{-(1/2)}\mathbf{h}^{(l-1)}\mathbf{W}^{(l-1)}\right). \tag{6}
$$

In the formula, $\sigma(\cdot)$ is the nonlinear activation function, and $\mathbf{w}^{(l)}$ is the l-th layer graph convolution of the weight matrix of the network.

After the concept of GCN was proposed, new forms of graph convolutional network models based on spectral-domain have been proposed, such as AGCN [16], AGC [17], and so on. However, the GCN based on the spectral domain cannot handle directed graphs and has poor scalability, while the GCN based on the spatial domain is more flexible and general. Spatial domain-based graph convolutional networks define graph convolutions according to the spatial relationships of nodes. NN4G [17] is the earliest proposed GCN based on spatial domain, which realizes graph convolution by directly accumulating feature information of node neighborhood. The message passing neural network (MPNN) proposed by Gilmer et al. [18] can be regarded as a general framework for GCN based on the spatial domain. MPNN decomposes the spatial domain convolution into two processes: information transfer and state update, and it takes the feature of node v as the initial state of the hidden state, namely,

$$
\mathbf{h}_v^{(0)} = \mathbf{x}_v, \tag{7}
$$

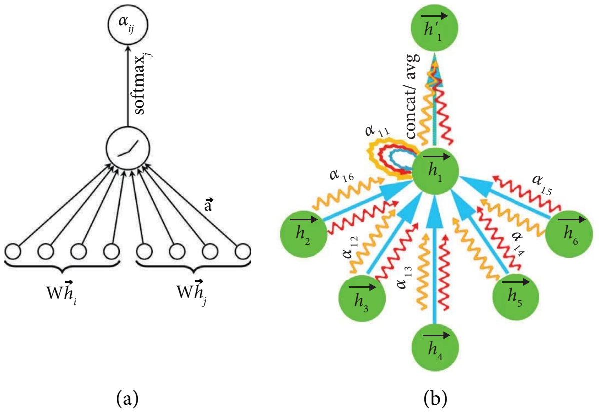
where $\mathbf{x}_v$ is the feature of node v. The hidden state update formula of MPNN is as follows:

$$
\mathbf{h}_v^{(l)} = U_l\left(\mathbf{h}_v^{(l-1)}, \sum_{u\in N(v)} M_l\left(\mathbf{h}_v^{(l-1)}, \mathbf{h}_u^{(l-1)}, \mathbf{x}_{vu}^{e}\right)\right), \tag{8}
$$

where l is the layer index, $U_l(\cdot)$ is the update function, and $M_l(\cdot)$ is the information transfer function. After obtaining the hidden representation of all nodes in the graph, the representation of the entire graph can be generated by the readout function

$$
\widehat{\mathbf{y}} = R\left(\mathbf{h}_v^{(l)} \middle| v \in G\right), \tag{9}
$$

where R(·) is the readout function. By defining different forms of update function, information transfer function and readout function, MPNN can represent a variety of spatial domain-based graph convolutional networks. Typical spatial domain-based graph convolutional networks also include PATCHY-SAN [19], GraphSage [20], and Diffusion CNN [21].



(a)

(b)

Figure 4: (a) Attention mechanism and (b) multihead attention mechanism.

2.2. Graph Attention Network (GAT). GAT introduces an attention mechanism on the basis of GCN, which enables the model to focus on the information most relevant to the current task, thereby improving the performance of the model. In spectral domain-based GCN, the filter function depends on the Laplacian matrix, which is derived from the graph structure, which makes the model trained on a specific graph cannot be directly applied to other graph structures. To solve this problem, Velikovi et al. [22] proposed a new type of graph neural network structure, namely, GAT, and Figure 4 shows the attention mechanism of GAT.

The GAT learns the average value of the neighborhood features of each node in the graph and is diluted and weighted according to the importance of the neighborhood. The graph attention layer is the key structure of GAT to realize the attention mechanism. The graph attention layer takes the features of the nodes in the graph as input and outputs another set of higher-level node features that may have different cardinality. The graph attention layer realizes the conversion of input and output through the attention coefficient obtained by the attention mechanism a. The attention coefficient represents the importance of node j to node i, and its expression is as follows:

$$e_{ij} = a\left(\mathbf{W}\mathbf{x}_i, \mathbf{W}\mathbf{x}_j\right). \tag{10}$$

In the formula, W is the weight matrix applied to all nodes, representing the relationship between input features and output features; $\mathbf{x}_i$ and $\mathbf{x}_j$ are the features of node i and node j, respectively. The model introduces the attention mechanism into the graph structure by only calculating the attention coefficients of nodes and their neighbors without considering the structural information of the graph. To simplify operations and facilitate comparison, the attention coefficients are regularized and used to generate output features as follows:

$$\mathbf{x}'_i = \sigma\left(\sum_{j\in N(i)} \alpha_{ij}\mathbf{W}\mathbf{x}_j\right), \tag{11}$$

where $\sigma(\cdot)$ is the nonlinear activation function; αij is the regularized attention coefficient. GAT also introduces a multihead attention mechanism similar to the transformer architecture, which can perform parallel computing on adjacent node pairs and stabilize the learning process. The complexity of the GAT method is low and it only pays attention to the adjacent nodes without the information of the whole graph, and it does not need to repeat the training model when it is applied to the new graph structure. For complex graph structures, some studies have proposed new graph attention networks, such as heterogeneous GAT [23], multirelational GAT [24], and spectral GAT [25]. These models can be used in more complex and informative networks to achieve better results.

2.3. Graph Autoencoder. GAE is an unsupervised learning framework that can convert graph structures into low-dimensional vectors and reconstruct graph structures using encoded information and is often used for graph embedding (GE) and graph structure generation [26]. Graph embedding is a graph representation learning (GRL) method that aims to map graph structure data into low-dimensional dense vectors while preserving node information. Graph embedding enables graph-structured data to be more efficiently applied to traditional machine learning algorithms to achieve better results in tasks such as recommendation and classification. Typical methods include random walk-based graph embeddings, such as DeepWalk [27] and Node2Vec [28] (Figure 5 shows the difference between DeepWalk and Node2Vec), and graph embeddings based on matrix decomposition, such as singular value decomposition (SVD), locally linear embedding (LLE), and non-negative matrix factorization (NMF). Compared with graph embeddings based on the random walk and matrix factorization, graph autoencoders can be applied to highly nonlinear graph structures, preserving the nonlinear structure and complex features of graphs. In 2014, Tian et al. [29] applied autoencoders to graph data for the first time, which took the adjacency matrix of the graph or its variants as the original node features and generated it by stacking sparse autoencoders (SAE). The nonlinear embedding of the graph, that is, the low-dimensional node representation. Structural deep network embedding (SDNE) [30] is an important graph autoencoder model that also adopts the stacked autoencoder structure. It maintains the local network structure and global network structure of the graph through the first-order similarity and second-order similarity between nodes, respectively. Multilayer nonlinear functions generate-graph embedding vectors. The hidden layer expression of SDNE is as follows:

$$\begin{aligned}\mathbf{h}_{v}^{(l)} &= \sigma\left(\mathbf{W}^{(l)}\mathbf{x}_{v} + \mathbf{b}^{(l)}\right),\\ \mathbf{h}_{v}^{(l)} &= \sigma\left(\mathbf{W}^{(l)}\mathbf{h}_{v}^{(l-1)} + \mathbf{b}^{(l)}\right),\end{aligned} \tag{12}$$

where $\mathbf{x}_v$ is the feature of node v, $\mathbf{W}^{(l)}$ is the weight matrix of the lth layer, and $\mathbf{b}^{(l)}$ is the deviation of the lth layer. After the final hidden layer output is obtained, the output representation x' can be obtained by inverting the calculation process of the encoder. SDNE contains two loss functions, of which the first loss function adopts the idea of a Laplacian feature map to preserve the first-order similarity, and its expression is as follows:

$$L_{1\text{st}} = \sum_{i,j=1}^{n} s_{i,j}\| \mathbf{h}_i - \mathbf{h}_j \|_2^2. \tag{13}$$

In the formula, $s_{i,j}$ represents the connection relationship of the nodes in the graph, if and only when the node i is connected with the node j, $s_{i,j} > 0$. The second loss function is used to maintain the second-order similarity, and a penalty vector is introduced to impose a larger penalty on the reconstruction error of nonzero elements than zero elements. Its expression is as follows:

$$L_{2\text{nd}} = \sum_{i=1}^{n} \| \left\|\left(\mathbf{x}_i' - \mathbf{x}_i\right) \circ b_i\right\|_2^2, \tag{14}$$

where represents the Hadamard product. $b_i = \left\{b_{i,j}\right\}_{j=1}^{n}$, $s_{i,j} = 0$, $b_{i,j} = 1$, otherwise $b_{i,j} = \beta > 1$. To maintain both first- and second-order similarity, the joint loss of SDNE is as follows:

$$L_{\text{mix}} = L_{2\text{nd}} + \alpha L_{1\text{st}} + \nu L_{\text{reg}}, \tag{15}$$

where L_{reg} is the regularized L2 norm, which is used to prevent overfitting.

Another type of graph autoencoder uses variational autoencoders (VAE) [31] to implement graph embedding, which is an important generative model and can improve the generalization ability of the model. VGAE [32] applies the variational autoencoder to the graph structure, and its inference model, namely, the encoder, utilizes a 2-layer GCN structure, whose expression is as follows:

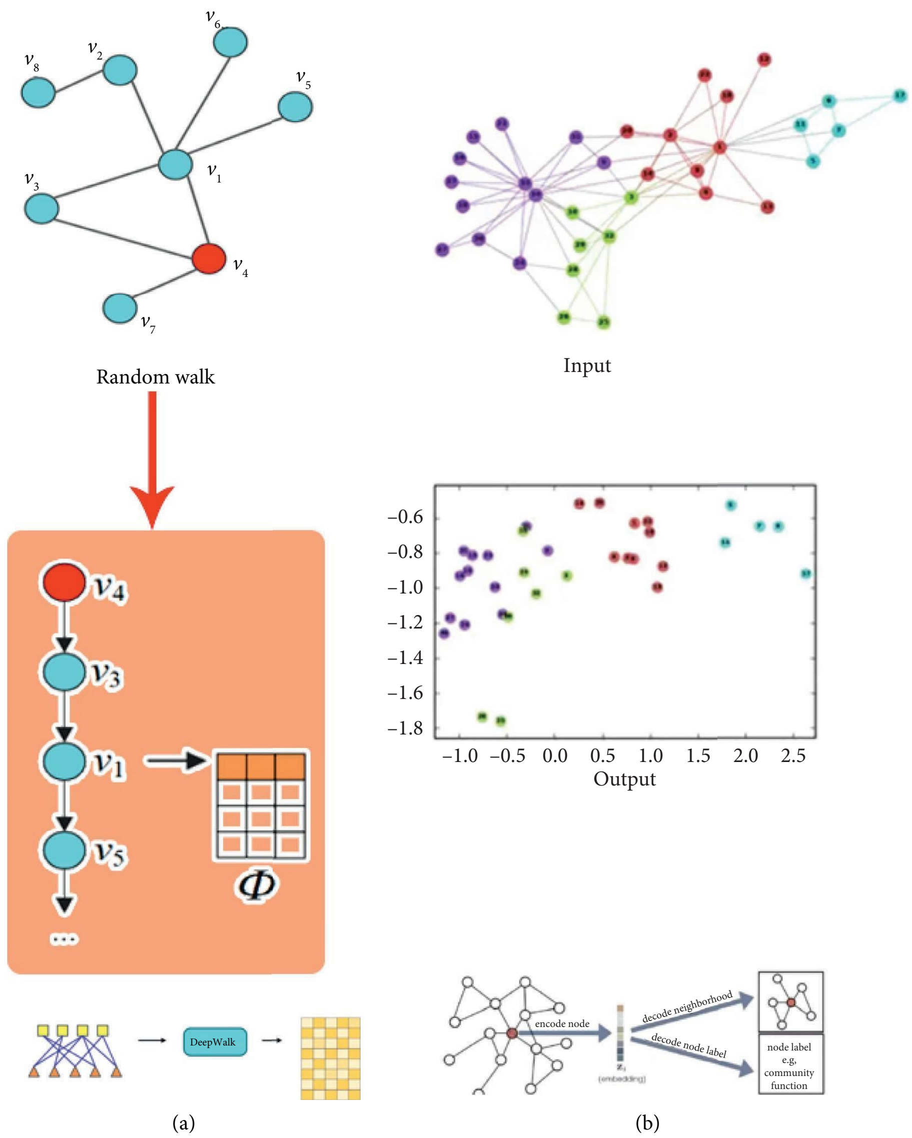
$$q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} q\left(\mathbf{z}_i \mid \mathbf{X}, \mathbf{A}\right), q\ \left(z_i\ \mathbf{X}, \mathbf{A}\right) = N\left(z_i \,\middle|\, \boldsymbol{\mu}_i, \text{diag}\left(\boldsymbol{\sigma}_i^2\right)\right), \tag{16}$$

where μ is the mean matrix of the encoder, $\log(\sigma)$ is the variance matrix, X is the feature matrix, A is the adjacency matrix, and z_i is the random latent variable. The generator function of VGAE, the decoder, is derived from the inner product of the hidden variables, and its expression is as follows:

$$p(\mathbf{A}|\mathbf{H}) = \prod_{i,j=1}^{N} \sigma\left(\mathbf{h}_i\mathbf{h}_j^{\mathrm{T}}\right). \tag{17}$$

2.4. Other Graph Neural Networks. In addition to graph convolutional networks and graph attention networks, commonly used graph neural networks also include gated graph neural networks (GGNNs) and spatial-temporal graph neural networks (STGNNs). The gated graph neural network is an improvement on the traditional graph neural network architecture. By introducing the gated recurrent unit (GRU) into the graph neural network, the performance of the model in the long-term propagation of information is improved. The gated graph sequence neural network proposed by Ruiz et al. [33], introduced the gated recurrent unit into the information dissemination process and controlled the iterative loop to a fixed number of steps and no longer needed parameter constraints to ensure convergence. In addition to this model, the gated graph neural network model also includes GAAN [34] and so on. The spatiotemporal graph [30] is a graph structure that depicts the interaction between entities in the spatial and temporal dimensions. It has three basic elements: nodes, spatiotemporal edges, and temporal edges. The feature matrix in the high-dimensional feature space will change with time. The spatiotemporal graph neural network can learn the hidden patterns in the spatiotemporal graph, and obtain the feature information of the time domain and the spatial domain in the graph structure at the same time. Spatiotemporal graph neural networks can be divided into methods based on recurrent neural networks (RNNs) and methods based on convolutional neural networks (CNNs). RNN-based STGNN captures spatiotemporal correlations through graph convolution. Compared with the RNN-based method, the CNN-based STGNN processes the spatiotemporal graph in a nonrecursive way, which can perform parallel computation and avoid the problem of gradient explosion or gradient disappearance, such as CGCN.

2.4.1. Graph Convolutional Neural Network Spectral Method. The lack of translation invariance on graphs makes it difficult to define convolutional neural networks in the node domain. The spectral method uses the convolution theorem to define graph convolution from the spectral domain. We first give some background on the convolution theorem.



(a)

(b)

Figure 5: (a) DeepWalk and (b) Node2Vec.

(1) Graph signal processing:

Convolution theorem: the Fourier transform of the signal convolution is equivalent to the product of the signal Fourier transform [35]as follows:

$$F(f \times g) = F(f) \cdot F(g). \tag{18}$$

Among them, *f*, g represents the two original signals, F (f) represents the Fourier transform of *f*, · represents the product operator, and ∗ represents the convolution operator. Perform inverse Fourier transform on both sides of (1), we can get

$$f * g = F^{-1}(F(f) \cdot F(g)). \tag{19}$$

Among them, $\mathrm{F}^{-1}(\mathrm{f})$ represents the inverse Fourier transform of the signal *f*. Using the convolution theorem, we can multiply the signal in the spectral space, and then use the inverse Fourier transform to convert the signal to the original space to realize the graph convolution, thus avoiding the convolution caused by the graph data not satisfying the translation invariance. Define difficult problems. The Fourier transform on the graph depends on the Laplacian matrix on the graph. In the following, we will give the definition of the Fourier transform on the graph.

The definition of the Fourier transform on the graph depends on the eigenvectors of the Laplace matrix. Taking the eigenvectors as a set of bases in the spectral space, the Fourier transform of the signal *x* on the graph is as follows:

$$\widehat{x} = U^{\mathrm{T}} x. \tag{20}$$

Among them, *x* refers to the original representation of the signal in the node domain. $\widehat{x}$ refers to the representation of the signal, *x* transformed into the spectral domain, and U^T represents the transposition of the eigenvector matrix, which is used for Fourier transform. The inverse Fourier transform of the signal x is as follows:

$$x = U\hat{x}. \tag{21}$$

Using the Fourier transform and inverse transform on the graph, we can implement the graph convolution operator based on the convolution theorem as follows:

$$x_G^* y = U\left(\left(U^T x\right) \odot \left(U^T y\right)\right). \tag{22}$$

Among them, $\overset{\star}{G}$ represents the graph convolution operator, x, y represents the signal of the node domain on the graph, and $\odot$ represents the Hadamard multiplication, which represents the multiplication of the corresponding elements of the two vectors. We replace the vector $U^T y$ with a diagonal matrix g_θ, then Hadamard multiplication can be transformed into matrix multiplication.

(2) Graph convolutional neural network based on convolution theorem.

Spectral convolutional neural network (Spectral CNN) [36] is the earliest method to construct a convolutional neural network on the graph. This method uses the convolution theorem to define the graph convolution operator at each layer and passes the gradient under the guidance of the loss function. Back-pass learning convolution kernel, and stack multiple layers to form a neural network. The structure of the m*th* layer of the spectral convolutional neural network is as follows:

$$X_j^{m+1} = h\left(U \sum_{i=1} F_{i,j}^m U^\top X_i^m\right), j = 1, \cdots, q. \tag{23}$$

Among them, p, q are the dimensions of the input feature and output feature, respectively, $X_i^m \in R^n$ represents the ith input feature of the node in the m*th* layer on the graph, $F_{i,j}^m$ represents the convolution kernel in the spectral space, and h represents the nonlinear activation function. In the spectral convolutional neural network, such a layer structure transforms the features from p-dimensional to q-dimensional, and based on the convolution theorem, graph convolution is realized by learning the convolution kernel.

The spectral convolutional neural network applies the convolution kernel to the input signal in the spectral space, and uses the convolution theorem to realize graph convolution to complete the information aggregation between nodes, and then applies the nonlinear activation function to the aggregation result and Stack multiple layers to form a neural network. The model does not satisfy the locality, so the locality of the spectral convolutional neural network is not guaranteed; that is, the nodes that generate information aggregation are not necessarily adjacent nodes.

The original intention of modeling a GCN is to use the graph structure to describe the information aggregation of adjacent nodes, and the spectral convolutional neural network introduced previously does not satisfy locality. Recently, graph wavelet neural network (GWNN) [37] proposed to use of the wavelet transform instead of Fourier transform to realize the convolution theorem.

The wavelet neural network pointed out that, similar to the Fourier transform, the wavelet transform also defines a method to transform the signal from the nodal domain to the spectral domain. Here, we use $\psi_s = \{\psi_{s1} \cdot \psi_{s2}, \cdots, \psi_{sn}\}$ to represent the basis of wavelet transform, where ψ_{si} represents the energy diffusion from the i*th* node, which describes the local structure of the i*th* node. The definition of the wavelet basis depends on the eigenvectors of the Laplace matrix, namely, $\psi_s = UG_sU^T$, where $G_s = \text{diag }\left(\{g_s(\lambda_i)\}_{i=1}^n\right)$, and the diagonal elements are obtained by applying the g function to the eigenvalues. Different g functions endow the wavelet base with different properties. In the wavelet neural network, the author uses the thermal kernel function, $g_s(\lambda_i) = e^{s\lambda_i}$.

Taking ψ_s as the base of the spectral space, the transformation matrix of the inverse wavelet transform in the figure is $\Psi_s^{-1} = UG_sU^T$, where G-s represent the replacement of the previous g function with $g_{-s}(\lambda_i) = e^{s\lambda_i}$.

Compared with the Fourier transform, the basis of the wavelet transform has several good properties: (1) the basis of the wavelet transform can be obtained by the Chebyshev polynomial approximation, avoiding the high cost of the Laplace matrix eigen decomposition, (2) the wavelet the transformed basis has locality, and (3) the locality of the wavelet basis makes the wavelet transform matrix very sparse, which greatly reduces the computational complexity of $\Psi_s^{-1}x$ and makes the calculation process more efficient. The parameter s is used to represent the range of heat diffusion, and it can be flexibly adapted to different task scenarios by adjusting the hyperparameters.

Using the wavelet transform on the graph to replace the Fourier transform, the m*th* layer structure of the wavelet neural network is defined as follows:

$$X_j^{m+1} = h\left(\Psi_s \sum_{i=1}^{p} F_{i,j}^m \Psi_s^{-1} X_i^m\right), j = 1, \cdots, q. \tag{24}$$

Compared with the spectral convolutional neural network, the wavelet neural network replaces the Fourier transform with the wavelet transform; that is, it replaces U and $\mathbf{U}^T$ with Ψ and Ψ_s^{-1}. Under such a set of wavelet bases, the graph convolutional neural network satisfies locality, and the computational complexity of the graph convolutional neural network is greatly reduced due to the accelerated computation and sparseness of the wavelet base. In addition to wavelet neural networks, there are also some works dedicated to realizing locality and accelerated computation of graph convolutional neural networks, but different from the way wavelet neural networks replace the substrate, these works achieve locality by parameterizing convolution kernels, while reducing parameter complexity and computational complexity.

GraphHeat [38] analysed the previous spectral methods from the perspective of filters and pointed out that spectral convolutional neural networks are nonparametric filters, while Chebyshev networks and first-order graph convolutional neural networks are high-pass filters. However, this is inconsistent with the smoothness prior to the task of graph semisupervised learning. Based on this, the graph thermal kernel network uses the thermal kernel function to parameterize the convolution kernel and then implements a low-pass filter.

2.4.2. Spatial Method of Graph Convolutional Neural Network. In contrast to the previous methods, which all start from the convolution theorem to define graph convolution in the spectral domain, the spatial method aims to start from the node domain and aggregate each central node and its adjacent nodes by deriving an aggregation function from each central node. The definition of the general framework draws attention to the fundamental problems of graph convolutional networks and provides a platform for a comparative analysis of previously published work in the field. Two recent papers aim to define a general framework for graph convolutional networks by combining their respective contributions. In particular, hybrid convolutional networks (MoNet) [20] concentrate on the lack of translation invariance on graphs and map the local structure of each node by defining a mapping function, which is then applied to each node. Message propagation networks (MPNNs) [18], on the other hand, are based on the aggregation of information propagation between nodes and propose a framework by defining a general form of the aggregation function. The lack of translation invariance makes it difficult to define graph convolutional neural networks, which is a necessary but not sufficient condition. When applied to a graph, hybrid convolutional networks define an orthogonal coordinate system and represent the relationship between nodes as a low-dimensional vector in the new orthogonal coordinate system. At the same time, the hybrid convolutional network defines a set of weight functions that can be used to train the network. The weight function acts on all adjacent nodes centered on a node. The input is the relationship between nodes (a low-dimensional vector), and the output is a scalar value. With this cluster of weight functions, the hybrid convolutional network obtains a vector representation of the same size for each node as follows:

$$D_j(x)f = \sum_{y \in N(x)} w_j(u(x,y))f(y), j = 1, \cdots, J. \tag{25}$$

Among them, N(x) represents the set of adjacent nodes of x, f (y) represents the value of the node y on the signal f, u(x, y) represents the node under the coordinate system u, the low-dimensional vector representation of the relationship, w_j represents The jth weight function, and J represents the number of weight functions. This step makes each node get a J-dimensional representation, and this representation integrates the local structure information of the node. The hybrid convolution model defines the shared convolution kernel on this J-dimensional representation

$$\left(f_G^* g\right)(x) = \sum_{j=1}^{j} g(j) D_j(x) f. \tag{26}$$

Different from the hybrid convolutional network, the message propagation network points out that the core of graph convolution is to define the aggregation function between nodes. Based on the aggregation function, each node can be represented as the information superposition of surrounding nodes and itself. Therefore, this model proposes a general framework for graph convolutional networks by defining a general aggregation function. The message dissemination network is divided into two steps. First, the aggregation function is applied to each node and its adjacent nodes to obtain the local structural expression of the node; then, the update function is applied to itself and the local structural expression to obtain the new express

$$m_x^{t+1} = \sum_{y \in N(x)} M_t\left(h_x^t, h_y^t, e_{x,y}\right), h_x^{t+1} = U_t\left(h_x^t, m_x^{t+1}\right). \tag{27}$$

Among them, h_x^t represents the hidden layer representation of the node x in the t-th step, $e_{x,y}$ represents the edge feature of the nodes x, y, M_t represents the aggregation function in the t-th step, m_x^{t+1} represents the local structure expression obtained by the node x after passing the aggregation function, and U_t represents the update function of the t-th step. Using the previously mentioned aggregation function and update function to designate each layer of the neural network, each node can continuously update itself with information from its own and neighboring nodes as the source information, and then obtain a new expression that is dependent on the local structure of the node. Some methods, such as those described previously, no longer rely on Laplacian matrices but instead design neural networks for learning aggregation functions in the context of the spatial framework. Aggregate functions learned through these methods can be tailored to specific tasks and graph structures, resulting in greater adaptability and flexibility. Different methods of Graph approach are summarized below in Table 1:

3. Applications of GCN

Researchers have been paying close attention to the graph convolutional neural network since it was first proposed, particularly in the fields of network analysis, recommender systems, biochemistry, traffic prediction, computer vision, and natural language processing, among others. It is not only traditional machine learning fields such as computer science, artificial intelligence, and signal processing that can benefit from graph convolutional neural networks but also interdisciplinary research fields such as physics, biology, chemistry, and the social sciences that can benefit from this technology. Different fields contain a variety of different graph data, and the relationship between nodes and edges is also different. How to combine domain knowledge to model the given graph data using GCN is a key issue in the application of GCN.

Table 1: Different graph-based approaches.

Method	Description
GAE	GAE is an unsupervised learning framework that can convert graph structures into low-dimensional vectors and reconstruct graph structures using encoded information and is often used for graph embedding (GE) and graph structure generation [26]. Graph embedding is a graph representation learning (GRL) method that aims to map graph structure data into low-dimensional dense vectors while preserving node information. Graph embedding enables graph-structured data to be more efficiently applied to traditional machine learning algorithms to achieve better results in tasks such as recommendation and classification
GAT	GAT introduces an attention mechanism on the basis of GCN, which enables the model to focus on the information most relevant to the current task, thereby improving the performance of the model. In spectral domain-based GCN, the filter function depends on the Laplacian matrix, which is derived from the graph structure, which makes the model trained on a specific graph cannot be directly applied to other graph structures
GCN	GCN introduces convolution operation into graph structure and is one of the most important GNNs at present. According to the different feature extraction methods, it can be divided into GCN based on spectral-domain and graph convolution network based on the spatial domain
STGNN	RNN-based STGNN captures spatiotemporal correlations through graph convolution. Compared with the RNN-based method, the CNN-based STGNN processes the spatiotemporal graph in a nonrecursive way, which can perform parallel computation and avoid the problem of gradient explosion or gradient disappearance, such as CGCN

3.1. GCN in Communication Network. The previous graph neural network method has the ability to deal with complex communication network problems and has been applied to network function virtualization, wireless network resource allocation, network modeling, and performance analysis. Software-defined network (SDN) and network functions virtualization (NFV) have been researching hotspots in the field of communication networks in recent years. SDN separates the control plane and forwarding plane of the network.

Obtain the topology and resource information of the entire network. NFV uses virtualization technology to separate network functions from traditional hardware devices, which improves the flexibility of network configuration. GNN can be used to solve problems in SDN and NFV that need to explore graph structures, such as dynamic resource allocation, service function chain (SFC) establishment, and virtual network embedding (VNE). Rafiq et al. proposed a supervised learning method for SFC traffic prediction. The method uses GNN to map the input historical traffic to the output predicted traffic and adjust the resource allocation accordingly. The graph neural network of this model trains 2 functions: the transfer function of the point and the output function. The transition function of node *n* inputs the features of *n*, all adjacent edge features, all adjacent node features and states, and outputs the state of node *n*. The output function computes nodes based on the state and characteristics of the point's output [39]. Li et al. [40] used GNN to predict NFV resource requirements so as to obtain advanced information about upcoming requests and improve the effectiveness of SFC reconstruction algorithms based on deep reinforcement learning. Network traffic migration is also an important branch of dynamic resource allocation. Sun et al. [41] proposed a method for NFV network traffic migration using GNN and deep reinforcement learning. The method maps the input network topology to the output network topology after migration, which is used to realize the expansion of network traffic, reduction, and load balancing. The essence of the SFC dynamic resource allocation problem is the transformation of the topology structure, the optimization goal is the total end-to-end delay, and there are no complex constraints, so it is easy to use GNN to solve.

The virtual network mapping problem is similar to the SFC establishment problem, but the network request and resource constraints are more complicated. VNE problems are divided into node mapping and link mapping. The existing methods of GNN to solve the VNE problem mainly focus on the node mapping. Habibi et al. [42] proposed a method of using GAE to assist VNE physical node classification. The input of the model is the adjacency matrix and the resource feature matrix, and a supervised learning model that can reconstruct the network topology is trained through the graph neural network. Yan et al. [43] proposed to use of GCN combined with deep reinforcement learning to complete the node classification task. This method uses actor-critic reinforcement learning, in which GCN is used to extract physical node features, and the features extracted from physical nodes and virtual network requests are fused through feed-forward neural networks (FF), and finally, the probability of mapping nodes is obtained. In fact, for the SFC establishment and VNE problems of large-scale complex networks, considering the complexity of node and link resources and optimization objectives, graph neural networks are powerful tools for extracting topological information, with the potential to provide faster and more optimized solutions.

With the rapid development and application of technologies such as 5G, the Internet of Things, and edge computing, the problem of resource allocation in wireless networks has become more and more important. Through the effective allocation of resources, various optimization goals can be achieved in different application scenarios, and the utilization rate of network resources can be improved. The problem with wireless power control is how to determine the transmit power of each transmitter so that the network can achieve the overall optimal signal-to-noise ratio. Its basic model is an optimization problem with constraints. The optimization target is the weighted sum of the ratio of signal to interference plus noise, and the constraint is the transmit power of the base station or equipment. Shen et al. [44] proposed to represent the multiuser wireless channel with a complete graph and used GNN to solve the power control problem. The node of the complete graph is a transceiver pair, and the node features include the direct channel state and weight; the link of the graph is the interference channel, and the link feature is the interference channel state. The method trains the transfer function and output function through GCN to output the optimal transmit power of each transmitter. Considering the situation of base stations and users in practical problems, Guo and Yang [45] proposed a method to solve the power control problem in heterogeneous networks. The nodes of this model include two kinds of heterogeneous nodes: base station and user. Heterogeneous nodes use different transfer functions and use parameter sharing to obtain output results. The wireless power control problem is not an intuitive graph structure problem, so it is necessary to transform the problem into a graph structure through modeling, and then use the GNN model to solve it.

Nakashima et al. [46] used GCN based on deep reinforcement learning to extract the features of the channel vector with topological information, and then generate the channel deployment strategy. This method can perform channel allocation in densely deployed wireless local area networks, thereby improving system throughput. Yan et al. [43] proposed an energy-saving topology control algorithm based on GCN, which uses GCN to imitate the maximum spanning tree algorithm for link prediction, and introduces new edges into the topology according to the probability graph, which optimizes the wireless network in 5G and B5G environments.

Network modeling and performance analysis is a fundamental problem to realize an efficient communication network. As mentioned previously, GNN can be used for resource optimization of wired and wireless networks. Various resources in the network are allocated to devices through optimization strategies. Therefore, an efficient network model is urgently needed to evaluate the quality of resource allocation. Rusek et al. [47] proposed RouteNet, which uses GNN to accurately evaluate the end-to-end delay and packet loss of network paths. RoutNet takes network topology, traffic matrix, and end-to-end path as input, and outputs performance evaluation indicators (delay, jitter, packet loss, etc.) according to the network state. RouteNet contains a multilayer information transfer neural network, uses RNN as the transfer function, compresses the link and path information into the hidden state vector, and finally obtains the evaluation index value of the path through the output function. RouteNet is used for the following two example problems: (1) routing optimization based on network delay and packet loss and (2) network topology upgrade with budget constraints.

Routing is an ancient and core optimization problem in the field of communication networks. Artificial intelligence algorithms have been used for network routing. Geyer et al. [48] proposed to use of GNN to learn distributed routing algorithm. This method abstracts the router interface as a point in the topology and uses GNN to train the hidden node information so that each node has a local representation of the graph topology. This method is a rare distributed-oriented GNN application. Secure network communication relies heavily on encrypted network traffic, and it can help protect sensitive data and maintain its integrity. However, it obscures the data's characteristics, makes it more difficult to identify malicious traffic, and shields such activity from detection. Consequently, encryption alone cannot guarantee fundamental information security. It is also important to keep an eye out for suspicious activity by monitoring traffic. Traffic classification methods based on statistical features and graphs are currently the most widely used. The limitations of these two approaches make them unreliable when used to detect malicious traffic that is encrypts its contents. Prior to this, the external connections between the network flows were not considered at all in the analysis. The latter, on the other hand, is completely the opposite. GCN model called GCN-ETA is proposed by [49] which considers the statistical features (internal information) of network flows and the structural information (external connections) between them to identify malicious traffic. GCN-ETA has two parts: an improved GCN feature extractor and a decision tree classifier. It is possible to enhance the effectiveness and speed of detecting malicious encrypted traffic by modifying the traditional GCN, and this can be used as a model for the implementation of GCN in similar scenarios [50]. The design of poisoning-resistant graph neural networks is extremely difficult, and several attempts have been made in the past. Existing research attempts to reduce the negative impact of adversarial edges only with the poisoned graph, which is suboptimal because they fail to distinguish between adversarial and normal edges. Tang et al. developed PA-GNN relies on a penalized aggregation mechanism that directly restricts the negative impact of adversarial edges by assigning them lower attention coefficients [51]. Pan et al. proposed the traffic classification method using GCN and LSTM, which low labeled sample for model classification and getting better accuracy using GCN [52].

3.2. GCN in Medical Imaging. Coronavirus 2019 (COVID-19) disease chest computed tomography (CT) scans are typically derived from multiple datasets gathered from various medical centers, with images sampled using a variety of acquisition protocols. However, while combining datasets from multiple sites increases sample size, it is hampered by intercenter heterogeneity, which makes comparisons difficult. The following steps are proposed by Song et al. [53] for diagnosing COVID-19 using an augmented multicenter graph convolutional network to address this issue: (AM-GCN). Amgen's convolutional neural network (AM-GCN) extracts features from initial CT scans using a 3-D CNN, which is supplemented by a ghost module and a multitask framework to improve the network's performance. This study uses the extracted features to construct a multicenter graph that takes into account intercenter heterogeneity, as well as the disease status of training samples, as described in the following section. In addition, the AM-GCN algorithm employs an augmentation mechanism to increase the number of training samples, resulting in an augmented multicenter graph. This method achieved a mean accuracy of 97.76 percent based on 2223 COVID-19 subjects and 2221 normal controls from seven medical centers.

Given the high cost of exhaustively annotating 3D data, a more sustainable approach would be to develop diagnosis algorithms using only patient-level labels. Chen et al. [54] proposes the Instance Importance-aware GCN (I^2GCN) for multi-instance learning (MIL), motivated by the fact that 2D slices of 3D data exhibit explicit diagnostic efficacy. To be more precise, this study begins by calculating the instance importance of each slice for diagnosis using a preliminary MIL classifier, which is then used to promote the refined diagnosis branch. Create the instance importance-aware graph convolutional layer (I^2GCLayer) in the refined diagnosis branch to exploit complementary features in both importance-based and feature-based topologies. Additionally, to address the deficient supervision of 3D datasets, the importance-based subgraph augmentation (SGA) technique was proposed to effectively regularize framework training.

Zhu et al. [55] developed Interpretable Dynamic GCN (IDGCN) to enhance the performance of personalized alzheimer's disease diagnosis and to generate interpretable results. This is accomplished through the incorporation of interpretable feature leaning and dynamic graph leaning into a GCN architecture. More precisely, interpretable feature learning ensures that diagnosis results are interpretable, and preclassification ensures that selected features are classification-oriented. Additionally, by adjusting the similar and dissimilar correlations of all objects, dynamic graph learning dynamically updates the graph structure for GCN to produce superior diagnosis results. Thus, by optimizing feature learning, graph learning, and the GCN simultaneously, the proposed disease diagnosis method not only generates reliable personalized diagnoses but also provides interpretability for diagnosis results. Similarly, Jiang et al. [56] proposed a hierarchical GCN framework (called hi-GCN) to learn the graph feature embedding, while considering the network topology information and subject's association at the same time. Memory, thinking, behavior, and emotion are all affected by dementia, which is a term used to describe progressive brain syndromes. A dementia patient's ability to carry out everyday tasks may deteriorate, and they become increasingly dependent on their caregivers. As a result, spotting the early signs of cognitive decline and alerting caregivers and doctors would be beneficial. Arfoglu et al. [57] used GCN to recognize activities and flag abnormal behavior related to dementia.

Figure 6 shows a standardized approach implemented by different studies for the classification of medical images. For medication recommendation and lab test imputation, Mao et al. [58] developed MedGCN, a machine learning framework based on MedGraph. The framework can be applied to a wide range of medical procedures. MedGCN built a graph to associate four different types of medical entities, namely patients, encounters, lab tests, and medications, and then used a graph neural network to learn node embeddings for medication recommendation and lab test imputation. Shi et al. [59] proposed a new method called cell-graph convolutional neural network (CGC-Net) that converts each large histology image into a graph in which each node is represented by a nucleus within the original image and cellular interactions are denoted as edges between these nodes based on node similarity. To improve the algorithm's performance, the CGC-Net employs nuclear appearance features in addition to spatial node location. Zhang et al. [60] proposed a BDR-CNN-GCN model using batch normalization with CNN and GCN to get an accurate classification of breast disease. Yin et al. [61] create a novel multi-instance deep learning method for building a robust classifier by treating multiple 2D ultrasound images of each individual subject as multiple instances of the same bag. Convolutional neural networks (CNNs) are used in this method to learn instance-level features from 2D US kidney images, and GCNs are used to further optimize the instance-level features by exploring potential correlations among instances of the same bag. This study also use full-connected neural networks (FCNs) to learn bag-level features using gated attention-based MIL pooling. Table 2 gives a detailed comparison of different approaches using graphs.

3.3. GCN for Recommendation and Prediction. In today's web platforms and applications, recommender systems are widely used as important tools to alleviate information overload and improve user experience. They are now widely deployed. It is a hot topic right now to take more user preferences into account when making recommendations. Although the systems often choose "click" or "purchase" as the optimization target in real-world information systems, there are also various types of user behaviors, such as view and add-to-cart. Users have the option of viewing, adding to a cart, and ultimately purchasing any item. In order to create a more precise recommender system, data on a user's diverse behaviors is crucial. Before this study, researchers would typically use a default value (i.e., "other") to represent the missing attribute, which resulted in suboptimal performance. In order to address this issue, Liu et al. propose an attribute-aware attentive graph convolution network that is both fast and accurate (A2-GCN) [20]. The A2-GCN algorithm, in particular, begins by constructing a graph in which nodes represent users, and items represent attributes.

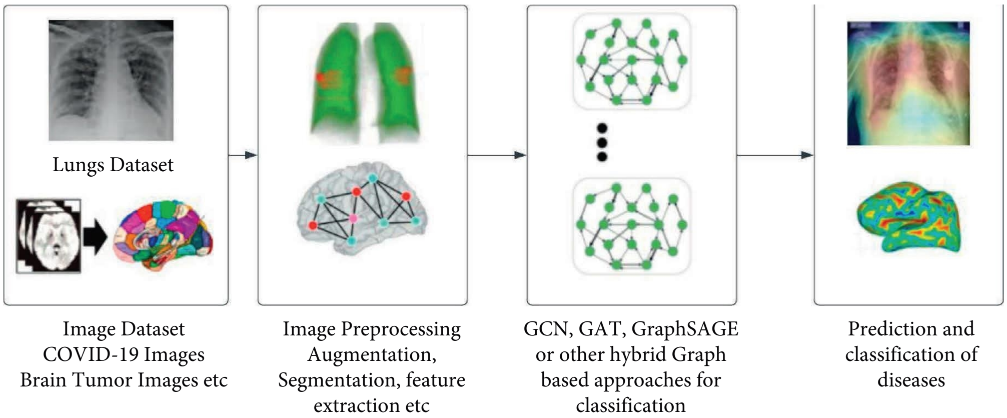


Figure 6: Medical image classification using GCN.

Following that, A2-GCN makes use of the graph convolution network to characterize the complex interactions among the participants. This model also employs the message-passing strategy to aggregate the messages passed from the other types of nodes that are directly linked in order to learn the node representation (see Figure 2) (e.g., a user or an attribute). Guo et al. used a similar approach and developed a domain-aware GCN (DA-GCN) model, which links users and items in each domain as a graph [67]. Shehnepoor et al. used GCN for a recommendation of fraudster detection in rating the user profile and proposed a HIR-RNN algorithm [68]. This algorithm performs two tasks i.e., prediction of user rating and fraudster detection based on user behavior.

Knowledge graphs (KGs) when combined with a recommendation system are helpful for providing the explainable recommendation. Ma et al. proposed knowledge-aware reasoning with graph convolution network (KR-GCN) which integrates user-item interactions and knowledge graphs into a heterogeneous graph, which is performed with the GCN [69]. Social data are much more important for getting a recommendation about the product, and Yu et al. proposed an enhanced social recommendation system based on GCN, which solves the problem of limited neighbors, noisy social relationships, and heterogeneous neighbors [70]. This model use an autoencoder to augment the data by encoding the high-order and complex connectivity patterns [71]. To extract the relationship between indirect instances between users and items, hamming similarity model is proposed by Liu et al. named hamming spatial graph convolutional networks (HS-GCNs). Xiao et al. proposed a GCN model for a recommendation system using a deep graph neural network named DeepFM graph convolutional network (DFM-GCN) [72]. DFM-GCN is mainly focused on solving the problem of cold start and data sparseness, which is solved by getting the interactive information between the nodes and the representation of items as vector nodes in GCN. There are many types of interaction data that can be generated by users, but traditional studies on recommender systems tend to focus on just one type of user behavior (the optimization target, for example, purchasing) (e.g., view, click, add-to-cart, and so on). Well-structured information can be derived from heterogeneous multi-relational data, and this information can be used to make excellent recommendations. As a result, early attempts to leverage these heterogeneous data fail to capture the high-hop structure of user-item interactions, which are insufficient to make full use of them and may only deliver limited recommendation performance. Graph heterogeneous collaborative filtering (GHCF) explores the high-hop heterogeneous user-item interactions; this study takes the advantages of graph convolutional network (GCN) and further improves it to jointly embed both representations of nodes (users and items) and relations for multirelational prediction data sparsity issue is further solved by Tang et al. by developing multigraph collaborative filtering (DMGCF) model to mine and reuse side information. This method generates multiple graphs with a dynamic evolution mechanism to simulate side information for better performance, especially when side information is unavailable [73].

Monti et al. [74] combined a multi-graph convolutional neural network with a recurrent neural network, in which the multigraph convolutional neural network was used to extract locally stationary features, and the recurrent neural network could diffuse the fractional values and reconstruct the matrix. Zhang et al. [75] modeled the recommender system as a link prediction problem on graphs and proposed a graph self-encoding framework based on different message propagation to model the bipartite graph of the recommender system and achieved the best results on data including social networks good result. Yang et al. [76] applied convolutional neural networks to recommender systems and proposed a data-efficient graph convolutional neural network algorithm MultiSage to generate embedded representations for commodity nodes. These expressions contain graph structure and node feature information. Compared with the traditional graph convolution method, it proposes an efficient random walk strategy to model convolution, designs a new training strategy, and successfully integrates graph convolution neural networks. The network is applied to a superlarge-scale recommendation system with 1 billion nodes. Wang et al. [77] proposed the RippleNet framework, which introduced knowledge graph information to improve the performance of the recommender system. Liao et al. [78] proposed the SocialLGN framework, which includes three

Table 2: Different methods for medical image classification using graphs.

Reference	Proposed method	Technique	Field of study	Dataset	Results
Song et al. [53]	AM-GCN	(i) 3D CNN for feature extraction (ii) GCN for classification of heterogeneous networks	CT images of COVID-19	(i) Wuhan Keting1 (ii) Wuhan Keting2 (iii) Harbin Medical University (iv) Wuhan Shelter	Accuracy > 97.96 Sensitivity > 98.58
Chen et al. [62]	Instance importance-aware graph convolutional network (I2GCN)	(i) GCN for 3D datasets (ii) Multi-instance learning(MIL)	3D CT images of COVID-19	CC-CCII	
Zhu et al. [55]	Interpretable dynamic graph convolutional networks (IDGCNs)	(i) Interpretable feature learning (ii) Dynamic GCN	Alzheimer’s disease (AD) MRI images	ADNI database	Accuracy 1.04% improved
Jiang et al. [56]	Hierarchical GCN (Hi-GCN)	(i) Multiple hierarchical GCN	Alzheimer’s disease (AD) MRI images	(i) Alzheimer’s disease neuroimaging initiative (ADNI) dataset (ii) Autism brain imaging data exchange (ABIDE)	Average accuracy > 73.1% AUC > 82.3%
Arifoglu et al. [57]	GCN	(i) Activity recognition using GCN	Dementia brain disorder	Self-data generation through simulation	
Yang et al. [63]	Gated GCN with attention convolutional binary neural Tree (GGAC)	(i) Attention mechanism (ii) Gate GCN	Chest X-rays		
Zhang et al. [64]	BDR-CNN-GCN combines two advanced neural networks: (i) graph convolutional network (GCN) and (ii) convolutional neural network (CNN)	(i) Batch normalization (BN) (ii) Dropout (DO) (iii) GCN (iv) CNN	Breast cancer	Mini-MIAS dataset	Sensitivity: 96.20 ± 2.90% Specificity: 96.00 ± 2.31% Accuracy: 96.10 ± 1.60%
Wu et al. [65]	CNN-GCN	(i) CNN for feature extraction (ii) GCN for prediction	Skin disease	Telehealth platform from a local hospital	Accuracy: 93.6%
Xuan et al. [66]	DGC: dynamic graph convolution	(i) Node attribute attention mechanism (ii) GCN-based autoencoder	Kidney diseases 3D images	2019 kidney tumor segmentation challenge dataset	Dice coefficient > 96.1% Intersection over union > 89.0% Hausdorff distance > 17.51

Table 3: Different methods for recommendation and prediction using graphs.

Reference	Technique	Focused area	Datasets	Performance
Liu et al. [20]	A2GCN: attribute-aware attentive graph convolution network	Relational attributes	(i) Amazon review dataset	Hit Ratio > 2.89% NDCG > 6.77%
Guo et al. [79]	DA-GCN: domain-aware attentive graph convolution network	Knowledge transfer via multiple relational entities	(i) HVIDEO (ii) HAMAZON	MRR@5 > 19.24 Recall @5 > 26.25
Zhang et al. [8]	GraphRf: GCN-based user representation learning framework	Fraud detection in an e-commerce application	(i) Yelp (ii) Movies and TV	Precision > 98% Recall > 99% F1 score > 97%
Ma et al. [69]	KR-GCN: knowledge-aware reasoning with graph convolution network	Knowledge-based recommendation	(i) Yelp2018 (ii) Amazon-Book (iii) Last-FM	NDCG > 5.7% improved Recall > 3.1% improved
Yu et al. [70]	Concatenate the motif-based GCN with a fully-connected multilayer perception (MLP)	GCN-based autoencoder	(i) LastFM (ii) Douban (iii) Gowalla	NDCG > 0.06 Recall > 4.80% Precision > 3.97%
Liu et al. [80]	HS-GCN: hamming spatial graph convolutional networks	Captures the first and high-order Hamming similarities then perform GCN	(i) MovieLens (ii) Yelp (iii) Amazon (iv) Netflix (v) Pinterest (vi) Gowalla	NDCG > 5.49 and <56.24 HR > 3.87% and <44.87%
Xiao et al. [72]	DFM-GCN: DeepFM is used to capture the interactive information GCN is for deep learning	Prediction of the click-through rate for recommendation	MovieLens dataset 2 versions: (i) Ml1m_kg1m (ii) ml1m_kg20m	Precision Recall F1

parts: user modeling, commodity modeling, and scoring prediction. Using the attention mechanism, the user's interaction information and the user's social network information are effectively modeled.

The graph convolutional neural network is considered to be able to model the structural attributes and node feature information of the graph well, and the recommendation system can be regarded as either a matrix completion problem or a bipartite graph (users and items) for the link prediction problem. Compared with traditional methods, graph convolutional neural networks can better utilize the information of user attributes and product attributes that are ubiquitous in recommender systems, which is why graph convolutional neural networks can attract widespread attention in recommender system tasks. Table 3 highlights the latest progress in recommendation and prediction using the graph-based methods.

3.4. GCN for Hyperspectral Data. Hyperspectral techniques have been greatly improved by the rapid growth of optics and spectroscopy. A considerable quantity of important information can be captured using hyperspectral images, which are images that contain many contiguous bands. It has been used in a variety of disciplines, including military target identification, vegetation monitoring, and disaster prevention and control throughout the past few decades [4]. Various algorithms have been proposed so far for categorizing the pixels of a hyperspectral image into specific land-cover categories. The early-stage approaches rely heavily on traditional pattern recognition methods such as K-nearest neighbor classifiers and linear classifiers. K-nearest neighbor has been frequently employed among these traditional methods due to its simplicity in both theory and practice. With high-dimensional hyperspectral data, support vector machine (SVM) works stably and satisfactorily. The aforementioned approaches, however, are all based on constructed spectral-spatial properties that rely largely on professional skill and are extremely empirical. Deep learning is being used to solve this flaw. Mou et al. were the first to use a recurrent neural network (RNN) to classify hyperspectral images [81]. Convolutional neural network (CNN) has recently emerged as a potent method for hyperspectral image classification, and Lou et al. developed a high-performance novel HSI classification algorithm based on CNN. Figure 7 shows the basic implication method in different studies.

It's been shown that convolutional neural networks (CNNs) are excellent at representing and classifying hyperspectral images. Convolution can only be performed on normal square image regions with fixed sizes and weights in traditional CNN models, so they cannot generically adapt to unique local regions with varying item densities and geometric appearances. They must be improved in classification, especially at class boundaries [82]. To address this shortcoming, Luo et al. [83] propose using the recently proposed graph convolutional network (GCN) for hyperspectral image classification, as it can perform convolution on arbitrarily structured nonEuclidean data and is applicable to irregular image regions represented by spatial graph information. Mou et al. [84] proposed a graph-based semisupervised network called nonlocal-GCN. Unlike existing CNNs and RNNs, which take pixels or patches of a hyperspectral picture as input, this network takes the entire image (including both labeled and unlabeled data) into account. To be more specific, a nonlocal graph is initially computed. To extract characteristics from this network representation, a pair of graph convolutional layers are used. Finally, the network's semisupervised learning is accomplished by employing a cross-entropy error over all labeled occurrences. Ding et al. [32] adopt graphSAGE for feature extraction in local regions of the graph, which is helpful in getting more accurate and effective information about nodes. Using MSAGE-CAL based attention method with graphSAGE improves the classification accuracy of HSI.

Guo et al. [79] found that GCN models are shallow and feature extraction is not effective. To solve this issue, DGU-HSI is proposed. This DGU-HSI constructs two separate graphs for spatial and spectral data for feature extraction, which extract features simultaneously. Once the feature is extracted, the graph u-nets is used for the fusion of features for classification. Yang et al. [85] also used a similar approach using the spectral (Se-GCN) and spatial (Sa-GCN) data to develop an adaptive cross-attention-driven spatial–spectral graph convolutional network (ACSS-GCN). This model is improved by using an attention mechanism in both the blocks of spectral and spatial information. Qu et al. [86] used first-time GCN for change detection in HSI data and proposed dual-branch difference amplification GCN (D2AGCN) which is highly efficient in low samples of data. The dual-branch structure can effectively extract sufficient different features to facilitate the detection of the changed areas.

GCN is used by the methods described previously to investigate large-range spatial relations of HSI, whereas local spatial information is more important when training samples are limited. S2RGANet (spectral–spatial residual graph attention network), a novel method for HSI classification that addresses the shortcomings mentioned previously, has been developed to address these issues. The spectral residual modules in the S2RGANet are designed to extract spectral discriminative features, while graph attention convolutions are introduced to explore the local geometric structure. In contrast to existing GNNs, which are designed to learn large-range spatial relations between samples in HSI, the proposed graph convolutions are designed to capture the distribution pattern of land cover in a given local patch of ground [86]. Sha et al. [87] used GAT for HSI classification by using different weights for different nodes according to their attention coefficients during the convolution process. Table 4 highlights the latest progress in hyperspectral data classification using the graph-based methods.

3.5. GCN for Computer Vision. With the development of science and technology, image processing technology has been presented to everyone's field of vision. Among them, the content covered is relatively extensive, and the image information can be optimized, and the corresponding image

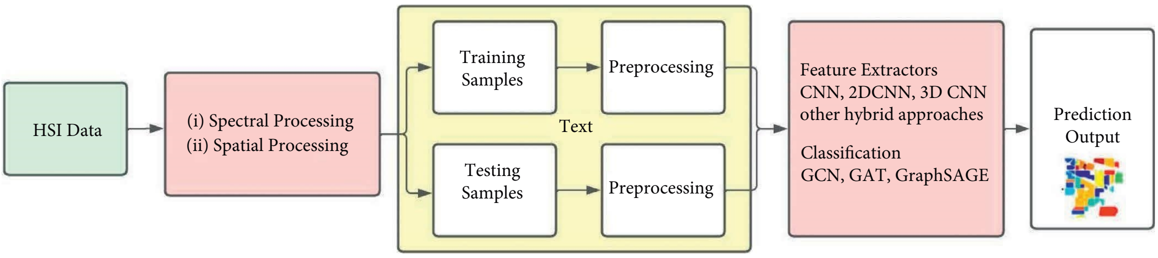


Figure 7: Hyperspectral image classification using GCN.

recognition, detection, data encoding processing, enhanced restoration, and other work can be completed. It can not only transmit the information that the people need to obtain but also penetrate into all aspects of work production. For example, the fields of transportation, agriculture, communication technology, and aviation all require the support of image processing technology. AI is playing a vital role in image processing and helping in different tasks of image processing efficiently such as image segmentation, change detection, denoising, image enhancement, and 3D images. After traditional approaches such as SVM, KNN, and so on. CNN provides a wide range of applications in image processing but currently, the research is diverting towards GCN because of better results and low complexity of calculation. GCN research in image processing is extensive in remote sensing images, medical images, 3D images, etc., with different types of images denoising and image enhancement techniques.

Saha et al. [90] used GCN for developing a change detection mechanism in remote sensing images. This semisupervised CD method encodes multitemporal images as a graph via multiscale parcel segmentation that effectively captures the spatial and spectral aspects of the multitemporal images. Ismail et al. [91] proposed a BLDNet algorithm for estimating the damage detection in the building caused by disasters or earthquakes. The model is based on Siamese CNN with trained GCN in semisupervised to get the predictions of disasters carried out by earthquakes. In recent years, deep learning-based image denoising methods have outperformed traditional denoising techniques. In order to train a convolutional neural network to infer clean images, most deep learning-based image denoising methods use cropped small patches. However, in practice, real-world noisy images tend to be of high resolution rather than the cropped small patches, and the vanilla training strategies ignore the cross-patch contextual dependency in the whole image. Li et al. [92] used cross-patch GCN with the help of CNN to perform denoising of the image, and the results show that denoising is 95% accurate. Shen et al. [93] extended the denoising work using GCNs and proposed a novel approach, GCN-Denoiser, which preserves features of mesh denoising and performs graph convolution operations in the dual space of mesh triangles.

Remote sensing (RS) image semantic segmentation, as the fundamental task of GIS, serves as the foundation for other RS research and applications such as natural resource protection, land cover mapping, and land-use change detection. Despite receiving significant attention over the last decade, semantic segmentation of high-resolution RS images remains difficult due to the complexity of structure in RS images, which leads to interclass similarity and intraclass variability. Ouyang et al. [94] proposed a DSSN-GCN framework that combines deep semantic segmentation with GCN. In this framework, an attention residual U-shaped network (AttResUNet) is used as a feature extractor while graph nodes are denoted by the superpixels, and the graph weight is calculated by considering the spectral information and spatial information. Kim et al. [95] proposed the Split-GCN model which outlines the objects in by similar nodes features and highlights them in a specified region. This model consists of two parts: an encoder (feature extraction network) to extract the boundary information of an object and a decoder (novel graph composition network) to capture the shape of an object. The model used the polygon-based approach to detect the object boundary in uniform spaced points.

Computer vision is a long-running research topic because it can perceive and recognize the world without the aid of humans by gathering data from sensors. Reverse engineering, intelligent surveillance, and remote sensing all rely on target recognition as a critical component of their respective applications. Unmanned systems and augmented reality are examples of practical application scenarios where three-dimensional (3D) object recognition is more relevant than two-dimensional (2D) target recognition. Zhan et al. [96] proposed a 3D point cloud model named minimum bounding box oversegmentation GCN (MBBOS-GCN). This model uses a minimum bounding box algorithm, and the farthest point sampling (FPS) algorithm is used to sample within each small region to reduce sampling randomness, with an accuracy of the model being more than 90% for segmentation of the 3D objects. Wang et al. [97] used GCN models for activity recognition in 3D space and proposed a spatial-temporal graph convolutional network (ST-GCN) model. In this method, semantically close point data are treated as neighbors, and using the motion capture data, a graph was created as follows: the intrabody edges between skeleton areas are defined based on the natural connections in human bodies; the interframe edges connect the same skeleton areas between consecutive frames. For 3D shape analysis, Wei et al. [98] proposed a model named View-GCN which can 3D shape based on a graph representation of multiple views inflexible view configuration. This model is a flexible model with different view configurations, e.g.,

Table 4: Different methods for hyperspectral data classification using graphs.

Study	Proposed method	Focused area	Technique	Dataset	Performance
Mou et al. [84]	Nonlocal-GCN	HSI classification	(i) Nonlocal graph is calculated (ii) GCN is performed	(i) Pavia University (ii) Indian Pines (iii) Salinas	OA > 87.04% AA > 90.87% Kappa > 86.26%
Ding et al. [88]	Multiscale GCN (MSAGE-CAL)	HSI classification	(i) GraphSage for feature extraction (ii) Context-aware learning	(i) Kennedy Space Center (ii) Pavia University (iii) Salinas	OA > 96.14% AA > 94.42% Kappa > 97.0%
Guo et al. [45]	DGU-HSI	HSI classification	(i) Spectral graph construction for feature extraction (ii) Spatial graph construction for feature extraction (iii) Applying fusion to data	(i) Pavia University (ii) Indian Pines (iii) Salinas (iv) Trento	OA > 96.62% AA > 95.32% Kappa > 94.0%
Yang et al. [63]	Adaptive cross attention-driven spatial–spectral graph convolutional network (ACSS-GCN)	HSI classification	(i) Spatial GCN (Sa-GCN) subnetwork (ii) Spectral GCN (Se-GCN) subnetwork (iii) Graph cross-attention fusion module (GCAFM)	(i) Pavia University (ii) Indian Pines	OA > 94.84% AA > 96.40% Kappa > 94.10%
Qu et al. [86]	Dual-branch difference amplification GCN (D2AGCN)	Change Detection		(i) China (ii) Bay Area (iii) Santa Barbara	OA > 94.74% Kappa > 85.68%
Xu et al. [89]	S2RGANet	HSI classification	(i) Spectral residual modules are employed to capture discriminant spectral features (ii) Graph attention mechanism for classification	(i) Pavia University (ii) Salinas	OA > 93.71% AA > 91.70% Kappa > 91.63%

Table 5: Different methods for image processing using graphs.

Reference	Algorithm	Focused area	Technique used	Dataset used
Saha et al. [90]	Semisupervised GCN	Change detection in VHR	(i) Using graph via multiscale parcel segmentation	Italy VHR dataset
Ismail et al. [91]	BLDNet	Damage detection by disaster	(i) Semisupervised GCN (ii) Neighbor feature extraction	Beirut Explosion dataset
Li et al. [40]	Cross-Patch Net (CPNet)	Image Denoising	(i) Patching for spatial and spectral data (ii) GCN denoising patches	(i) SIDD (ii) DND (iii) DIV2K
Shen et al. [93]	GCN-Denoiser	Image Denoising	Cascaded GCN	PrintData
Ouyang et al. [94]	DSSN-GCN	Image segmentation	Deep semantic segmentation network GCN	(i) UCM dataset (i) DeepGlobe dataset
Kim et al. [95]	Split GCN	Image segmentation	(i) ResNet-50 as image encoder (ii) Shape extraction from polygon objects using GCN	(i) Cityscapes dataset (ii) KITTI (iii) Rooftop (iv) ADE20 K (v) Card.MR
Zhan et al. [96]	MBBOS-GCN	3D segmentation	(i) Farthest point sampling (FPS) algorithm is used for the extraction of small samples (ii) GCN is used for semantic segmentation	(i) ModelNet40 (ii) ScanNet
Wang et al. [97]	ST-GCN	3D motion prediction	(i) Spatial information extraction (ii) Prediction of motion using GCN	(i) Nursing movement data
Wei et al. [98]	View-GCN	3D visualization of objects	(i) CNN for construction of features of graphs (ii) GCN for view generation	(i) ModelNet40 (ii) ShapeNet Core55 (iii) RGBD
Yao et al. [99]	Memory GCN (M-GCN)	Image recognition	(i) LSTM modules to interact with attention and prior knowledge (ii) GCN for deep semantic correlations	(i) Microsoft COCO (ii) PASCAL VOC
Nakao et al. [100]	Image to GCN (IGCN)	Image registration	(i) Performs image transition by a UNet-based network (ii) Vertex transformation by GCN	(i) 3D-CT volumes (ii) 4D-CT volumes

cameras located on circles, corners of dodecahedron, or even irregular positions around objects. Second, by using view-graph representation, this study can take advantage of GCN to aggregate multiview features considering the relations of graph nodes. Table 5 highlights some latest innovations for image processing using graphs.

3.6. GCN for Transport and Traffic System.

3.6. GCN for Transport and Traffic System. The traffic prediction problem is also one of the tasks in which graph convolutional neural networks are widely used. Its purpose is to predict the rate of future traffic given the historical traffic speed and route map. In the traffic prediction problem, nodes represent sensors placed on the road, and edges represent the physical distances of node pairs. Each node contains a temporal feature. Compared with traditional graph analysis problems, the traffic prediction problem includes both time and space modeling, and how to use the graph convolutional neural network to better model the road network in traffic brings opportunities and challenges. Li et al. [101] proposed a diffusion convolutional recurrent neural network (DCRNN) to model traffic forecasting. In this model, it regards traffic flow as a diffusion problem on a directed graph and proposes to use diffusion convolution to model graph-structured data. Use recurrent neural networks to model time dependencies. It achieves a 12%–15% improvement on two large-scale road network traffic datasets.

Cui et al. [102] proposed a traffic graph convolutional long short-term memory network (TGC-LSTM) to learn road networks and time-varying traffic patterns. It defines the graph convolutional neural network on the physical road network topology. The experimental results show that the method can capture the complex spatiotemporal dependencies effectively existing in the vehicle traffic network. Zhang et al. [1] proposed a graph gate recursive unit (GGRU) to solve the traffic flow prediction problem. They applied the graph gate recursive unit to the encoding-decoding model of the recurrent neural network and applied it to the Los Angeles highway data set. Zhang et al. [38] proposed a new deep learning framework, a space-time graph convolutional neural network (STGCN), to solve the problem of time series prediction in the traffic field. In this framework, it first formalizes the problem onto a graph using convolutional structures for modeling, which achieves significant improvements over traditional machine learning methods in short-term and mid-to long-term traffic prediction due to better utilization of topology.

Zhu et al. [103] developed AST-GCN for modeling traffic forecasting for intelligent transportation systems. This model uses external factors as dynamic attributes and static attributes and designs an attribute-augmented unit to encode and integrate those factors into the spatiotemporal graph convolution model and perform traffic speed prediction. In another approach, Zhu et al. [104] proposed BRB-based RNN-GCN model for traffic flow prediction, which solves the existing issues of traffic flow prediction models such as saturation or speed. In the scenarios related to traffic prediction, how to solve the spatiotemporal dependence is an important research direction. Since the graph convolutional neural network provides a solution to the modeling of graph data problems, it combines with time series models such as recurrent neural networks to give a good solution to the modeling traffic forecasting problem is presented [70]. However, how to further fine-grained consideration of spatiotemporal data modeling is still the focus of future research.

3.7. GCN for NLP. Graph convolutional neural networks have a large number of applications in the field of natural language processing. In this field, the more common graph data are knowledge graphs, syntactic dependency graphs and abstract meaning expression graphs, word cooccurrence graphs, and graphs constructed by other methods. Entity relation extraction (RE) is a method of encoding the meaning of a sentence as a rooted directed graph [105]. Sun et al. [106] applied graph convolutional neural networks to dependency syntax trees for machine translation tasks in English and German, and English and Zhou et al. [107] used a graph convolutional neural network for event extraction, where the graph used here is a dependency syntax tree. Table 6 provide a further description of the methods:

In addition to the previous graphs, word cooccurrence networks have also been applied to text classification tasks. Where nodes are nonstop words, and edges are word cooccurrence relationships in a given window. Defferard et al. [112] proposed a convolutional neural network defined in graph theory, which provides the necessary mathematical background and an efficient numerical scheme to design fast local convolutional filters on graphs. Reference [113] used a weighting approach with GCN for the categorization of text. Pal et al. [114] used graph convolutional neural networks for text classification tasks on the Reuters dataset. Yao et al. [115] applied a graph convolutional neural network to a text classification task by constructing a coword network and document relation network, and achieved the best results without using external knowledge and word representation.

A large number of studies have shown that the results of various natural language processing tasks have been improved to a certain extent after using the graph convolutional neural network model [116]. The use of graph structure enables the complex semantic relationship between objects to be effectively mined. Compared with the traditional serialization modeling for natural language processing, the use of graph convolutional neural networks can mine nonlinear complex semantic relationships.

3.8. GCN for Bioinformatics. In addition to the traditional modeling of graph data, graph convolutional neural networks have also received a lot of attention from researchers in fields such as biochemistry. Compared with traditional graph data research, in the field of biochemistry, people usually regard a chemical structure or a protein as a graph, the nodes in the graph are smaller molecules, and the edges

Table 6: Applications of graph network in NLP.

Method	Description
Threshold graph neural network	Hari et al. [108] used a threshold graph neural network (GGNN) on an abstract meaning graph for grammar-based machine translation tasks. Dependency syntax graph or tree, nodes are words, and edges are semantic relations
Sentence learning based on SRL	Song et al. [109] applied graph convolutional neural networks to reading comprehension, and the abstract meaning graph is applied to tasks such as text generation and relation extraction. The task of semantic role labeling (SRL) is to give a sentence and identify the predicate and the corresponding object in the sentence
GCN with LSTM	Marcheggiani et al. [110] proposed to use of a graph convolutional neural network to act on the syntactic dependency graph, and stack it with a long-short-term memory network
Dictionary-based learning using ST-GCN	Ye et al. [111] developed short text GCN (ST-GCN) in which different topic models for short text are employed, and a short text short-text graph based on the word cooccurrence, document word relations, and text topic information, is developed

represent bonds or interactions. Figure 8 is a molecular graph of medicine, where the nodes are carbon, hydrogen, and oxygen atoms, and the edges are chemical bonds. Researchers focus on the chemical function of a graph, that is, the object of study is no longer the nodes in the graph, but the entire graph itself.

Intracellular interactions are the focus of most methods for determining gene-gene interactions from expression data. High-throughput spatial expression data enables methods that can infer such interactions between cells and within cells. Yuan et al. [117] developed graph convolutional neural networks for Genes were developed to accomplish this (GCNG). It uses supervised training to combine spatial information with expression data. Prior methods for analyzing spatial transcriptomics data have been improved by GCNG, which can propose new extracellular interacting gene pairs. Upstream analyzes, such as functional gene assignment, can make use of the GCNG output. It is one of the primary goals of genomic medicine to identify the genetic variations in a patient that are responsible for their clinical phenotypes and to determine their relationship to those phenotypes. Only the genotype information is taken into consideration when prioritizing genomic variants, which results in the identification of a few hundred potential variants on average. It is still a significant challenge to narrow it down even further in order to identify the disease genes that are responsible for the clinical phenotypes observed. This is especially true for rare diseases. Motivated by the recent progress in spectral graph convolutions. Rao et al. [118] developed the graph convolution-based technique HANRD (Heterogeneous Association Network for Rare Diseases) to infer new phenotype-gene associations from this initial set of associations.

Predicting chemical compounds is one of the fundamental tasks in bioinformatics and cheminformatics because it contributes to various applications in metabolic engineering and drug discovery. Harada et al. [119] proposed a new graph convolutional neural network architecture called a dual graph convolutional network that learns compound representations from both the compound graphs and the intercompound network in an end-to-end manner. For the prediction of DNA protein, Zhang et al. [120] build a sequence k-mer graph for the whole dataset based on the k-mer cooccurrence and k-mer sequence relationship and then learn DNA graph convolutional network(DNA-GCN) for the whole dataset. It has not yet been thoroughly investigated whether advanced graph network methods can be used to identify functional protein complexes from the protein-protein interaction networks (PPIs) at a high level. To improve the detection of protein complexes, Zaki et al. [121] proposes a variety of graph convolutional network (GCN) methods. A node classification problem is first formulated as a protein complex detection problem. Once the model is developed and a complex affiliation matrix is in place, this model will be able to use it to group the nodes (proteins). In addition, a multiclass GCN feature extractor and a mean shift clustering algorithm are used to extract the nodes' features and perform clustering. Furthermore, applications are in Table 7:

Appropriate gene prioritization is critical for genome-based diagnostics of a variety of genetic diseases. However, it is a difficult task that requires a limited and noisy understanding of genes, diseases, and their associations. While several computational methods have been developed for the task of disease gene prioritization, their performance is largely constrained by manually crafted features, network topology, or predefined data fusion rules [127].

Li et al. [50] define convolutional neural networks directly on graphs. The neural network model inputs molecules of any size or shape and learns molecular fingerprints end-to-end. The model can better help to realize the molecular design of specific functions. Torng et al. [128] used a graph convolutional neural network to encode atoms, bonds, and distances, which can better utilize the information in the graph structure. It provides a new paradigm for ligand-based virtual screening. Gilmer et al. [129] proposed a message propagation model MPNNs to predict the chemical properties of a given molecule. Zitnik et al. [130] used graph convolutional neural networks to model multiple drug side effects. It first constructs multimodal maps of protein-protein interactions, drug-protein target interactions, and multiple drug interactions. In the graph, each side effect is treated as a different type of edge. Furthermore,

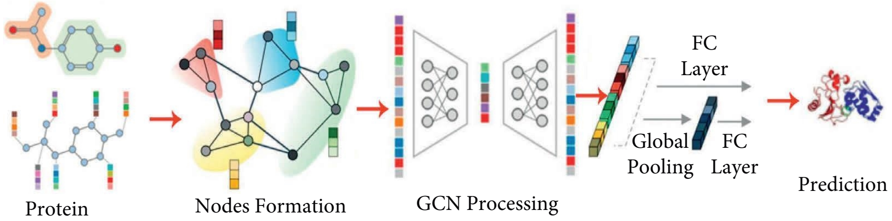


Figure 8: Protein prediction approach using GCN.

the modeling of side effects with drugs is transformed into a link prediction problem, which provides a new research idea for further study of pharmacology. Xiao et al. [131] and Reau et al. [132] proposed to apply graph convolutional neural network to protein interaction prediction. In this task, proteins are chains of amino acid residues that fold into three-dimensional structures that give them biochemical functions. Proteins exert their functions through complex networks of interactions with other proteins. You et al. [133] proposed the graph convolutional policy network (GCPN), a model based on general graph convolution and reinforcement learning to generate target graphs. In this model, the hidden state is expressed as a node by means of message propagation, and then a policy π is generated.

4. Prospects for Future Research Directions and Limitations of GCN

Although graph convolutional networks have succeeded in recent years, there are still some unsolved problems and directions worthy of further research.

4.1. Deep Network Structure. After stacking a large number of network layers, the traditional deep learning model has achieved remarkable results in many problems due to its powerful representation ability [38]. However, in the graph convolutional neural network model, after stacking a small number of layers, the network achieves the best effect. Adding graph convolutional layers simultaneously will make the results worse. Because graph convolution includes aggregating the features of neighbor nodes, when the network stacks multiple layers, the components between nodes are too smooth and lack discrimination. The experimental results of GCN show that when the number of network layers exceeds two layers, with the increase in the number of layers, the effect of GCN on the semisupervised node classification problem will decrease instead [134]. At the same time, with the continuous superposition of the network, eventually, all nodes will learn the same expression. Whether the graph neural network needs a deep structure or whether a deep network structure can be designed to avoid the problem of excessive smoothness is an urgent research problem to be solved.

4.2. Multiscale on-Graph Tasks. Graph mining tasks can be divided into node-level problems, graph and subgraph-level problems, and signal-level problems according to the different main objects. The critical point of node-level tasks is to learn efficient representations for each node, while learning representations for graphs is the key to graph-level tasks [135]. The key points of the signal-level task are learned to express effectively for different graph signals, while the network structure is unchanged. At present, most graph convolutional neural networks are designed for node-level tasks, and less attention is paid to graph-level and signal-level tasks.

4.3. Dynamically Changing Graph Data. In practical scenarios, the network is often dynamic. This dynamic includes the characteristics of nodes and edges that are constantly changing over time and the structure of the network that is constantly changing (there are new edges, nodes join the network, and nodes and edges disappear from the network) [136]. Considering the dynamics of the network is also the trend of graph mining algorithms. The current graph convolutional neural networks are designed for static networks, so designing a graph convolutional neural network that can model the dynamic changes of the network is also an important direction in the future.

4.4. The Complex Nature of Graph Data. In practical scenarios, networks often have complex structural characteristics. For example, the types of nodes, the complex features on the edges, the community structure of the network, and so on. Although many works have proposed some solutions, they are all networks designed for a certain characteristic [137]. Whether a network can be designed to simultaneously model various complex characteristics of the network is also a question worthy of discussion. GCN employs mean pooling. As a result, it will be impossible to distinguish aggregation on, say, the 2 different multisets (a,b) and (a,a,b,b). Mean-pooling will produce the same estimate for both multisets, so it is not injective. Because of mean-pooling, GCN will be unable to distinguish between nodes receiving messages from two other nodes and nodes receiving messages from four other nodes. The structural distinction is not distinguished here [133, 138–140].

4.5. Adversarial Attacks on Graph Neural Networks. Neural networks shine in various tasks but still have instability problems. For example, adding a certain amount of noise to the picture will not change the type of the picture to the human eye, but the neural network has already judged it as other types. Designing a targeted sample to make the

Table 7: Applications of graph network in bioinformatics.

Method	Applications
CADA	Peng et al. [122] proposes a novel graph convolutional network-based method for disease gene prioritization, CADA, based on the systematic embedding of the heterogeneous network formed by genes and diseases, as well as their unique characteristics
HPOFiller's	Liu et al. [123] proposed HPOFiller based on GCN for human phenotype ontology. HPOFiller's central idea is to repeatedly run these two GCN modules over the three networks in order to refine the embeddings. Extensive empirical evidence demonstrates that HPOFiller significantly outperforms all other state-of-the-art methods, including cross-validation and temporal validation, demonstrating that HPOFiller outperforms all other state-of-the-art methods. The ablation study demonstrates that batch normalization makes the greatest contribution to performance
Graph Attention Model (GAM)	Lee et al. [124] proposed a graph attention model (GAM), which adaptively selects some "information nodes" and collects the information of the entire graph for the problem of graph classification. Such et al. [125] directly defined the filter as a polynomial function of the graph adjacency matrix and proposed a Graph-CNNs model, which can handle heterogeneous and homogeneous graph data. On the molecular classification problem, it shows the best experimental results
MolGAN	Cao et al. [126] proposed MolGAN, which designs molecular structures containing specific chemical properties through generative adversarial networks combined with graph convolutional neural networks

machine learning model make a misjudgment is called an adversarial attack. In the field of GNN, constructing adversarial samples using the characteristics of nodes and network structure and designing a graph neural network that can defend against adversarial attacks are all important directions for future development.

5. Conclusion

Graphs are a powerful and rich structured data type with strengths and challenges that differ greatly from images and text. We have outlined some of the milestones that researchers have reached in developing neural network-based models that process graphs in this study. We have gone over some of the key design decisions that must be made when employing these architectures, and hopefully, the GNN playground can provide some insight into the empirical outcomes of these decisions. The recent success of GNNs opens the door to a wide range of new problems, and we are excited to see what the field will bring. The key points of this survey include the following points:

(i) The existing applications in the field of computer mainly use GN, GCN, and MPNN models, rarely use the GAE model and do not use the GAT model. Most of the existing applications use FF, RNN, CNN, etc., as aggregation functions to transmit node and topology information and output predicted values, and the application scope is limited. Due to their own limitations, GN, GCN, and MPNN are difficult to solve complex communication network problems.

(ii) The learning methods are mainly divided into supervised learning and reinforcement learning. Supervised learning is mostly used for traffic/resource/index prediction, node classification, and other issues; reinforcement learning is mostly used for path selection, topology transformation/mapping, and other issues.

(iii) The existing application goals mainly focus on the tasks of nodes. The output features are mostly the features of nodes or the overall indicators of the network and are rarely used for linking tasks.

(iv) Almost all existing applications are based on centralized learning, and it is necessary to obtain the information of all nodes before learning.

Consent

The authors allow to publish after acceptance.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Hao Tang, Uzair Aslam Bhatti, Guilu Wu, Shah Marjan, and Aamir Hussain were involved in the data analysis; Hao Tang, Uzair Aslam Bhatti, and Aamir Hussain were responsible for manuscript writing; Uzair Aslam Bhatti was responsible for project management; and Hao Tang was responsible for funding.

Acknowledgments

Hainan University Research Fund (project nos. KYQD (ZR)-22064, KYQD (ZR)-22063, and KYQD (ZR)-22065).

References

[1] C. Zhang and Y. Lu, "Study on artificial intelligence: the state of the art and future prospects," *Journal of Industrial Information Integration*, vol. 23, no. 2021, Article ID 100224, 2021.

[2] U. A. Bhatti, M. Huang, D. Wu, Y. Zhang, A. Mehmood, and H. Han, "Recommendation system using feature extraction and pattern recognition in clinical care systems," *Enterprise Information Systems*, vol. 13, no. 3, pp. 329–351, 2019.

[3] M. Abdar, F. Pourpanah, S. Hussain et al., "A review of uncertainty quantification in deep learning: techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021.

[4] U. A. Bhatti, Z. Yu, J. Chanussot et al., "Local similarity-based spatial–spectral fusion hyperspectral image classification with deep CNN and gabor filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[5] B. A. Miller, N. T. Bliss, and P. J. Wolfe, "Toward signal processing theory for graphs and non-Euclidean data," in *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5414–5417, IEEE, Dallas, TX, USA, March, 2010.

[6] M. M. Bronstein, J. Bruna, Y. Le Cun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[7] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 11–23, 2019.

[8] Z. Zhang, P. Cui, and W. Zhu, "Deep Learning on Graphs: A Survey," 2020, https://arxiv.org/abs/1812.04202.

[9] L. Alzubaidi, J. Zhang, A. J. Humaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, no. 1, pp. 1–74, 2021.

[10] D. L. Donoho and C. Grimes, "Image manifolds which are isometric to Euclidean space," *Journal of Mathematical Imaging and Vision*, vol. 23, no. 1, pp. 5–24, 2005.

[11] R. Angles and C. Gutierrez, "An introduction to graph data management," in *Graph Data Management*, pp. 1–32, Springer, Cham, Champa, 2018.

[12] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings of the 2005 IEEE international joint conference on neural networks*, vol. 2, pp. 729–734, Montreal, QC, Canada, July, 2005.

[13] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[14] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, https://arxiv.org/abs/1312.6203.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, https://arxiv.org/abs/1609.02907.

[16] C. Park, J. Park, and S. Park, "AGCN: attention-based graph convolutional networks for drug-drug interaction extraction," *Expert Systems with Applications*, vol. 159, Article ID 113538, 2020.

[17] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," 2019, https://arxiv.org/abs/1902.09130.

[18] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Message passing neural networks," in *Machine Learning Meets Quantum Physics*, pp. 199–214, Springer, Cham, Champa, 2020.

[19] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proceedings of the International conference on machine learning*, pp. 2014–2023, PMLR, New York City, NY, USA, June, 2016.

[20] S. Liu, J. H. Park, and S. Yoo, "Efficient and effective graph convolution networks," in *Proceedings of the 2020 SIAM International Conference on Data Mining*, pp. 388–396, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, June 2020.

[21] T. Mallick, P. Balaprakash, E. Rask, and J. Macfarlane, "Graph-partitioning-based diffusion convolutional recurrent neural network for large-scale traffic forecasting," *Transportation Research Record*, vol. 2674, no. 9, pp. 473–488, 2020.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, https://arxiv.org/abs/1710.10903.

[23] X. Wang, H. Ji, C. Shi et al., "Heterogeneous graph attention network," 2019, https://arxiv.org/abs/1903.07293.

[24] B. Li and Z. Zhu, "GNN-Based hierarchical deep reinforcement learning for NFV-oriented online resource orchestration in elastic optical DCIs," *Journal of Lightwave Technology*, vol. 40, no. 4, pp. 935–946, 2022.

[25] H. Chang, Y. Rong, T. Xu et al., "Spectral graph attention network," 2020, https://arxiv.org/abs/2003.07450.

[26] W. L. Hamilton, "Graph representation learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159, 2020.

[27] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," 2014, https://arxiv.org/abs/1403.6652.

[28] A. Grover and J. Leskovec, "node2vec: scalable feature learning for networks," 2016, https://arxiv.org/abs/1607.00653.

[29] F. Tian, B. Gao, Q. Cui, E. Chen, and T. Y. Liu, "Learning deep representations for graph clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, Laguna Hills, CA, USA, June 2014.

[30] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," 2016, https://arxiv.org/abs/1711.10146.

[31] C. Doersch, "Tutorial on variational autoencoders," 2016, https://arxiv.org/abs/1606.05908.

[32] Y. Ding, L. P. Tian, X. Lei, B. Liao, and F. X. Wu, "Variational graph auto-encoders for miRNA-disease association prediction," *Methods*, vol. 192, pp. 25–34, 2021.

[33] L. Ruiz, F. Gama, and A. Ribeiro, "Gated graph recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6303–6318, 2020.

[34] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D. Y. Yeung, "Gaan: gated attention networks for learning on large and spatiotemporal graphs," 2018, https://arxiv.org/abs/1803.07294.

[35] X. Zhi, D. Wei, and W. Zhang, "A generalized convolution theorem for the special affine Fourier transform and its application to filtering," *Optik*, vol. 127, no. 5, pp. 2613–2616, 2016.

[36] L. Yi, H. Su, X. Guo, and L. J. Guibas, "Syncspeccnn: synchronized spectral cnn for 3d shape segmentation," 2017, https://arxiv.org/abs/1612.00606.

[37] W. Zheng, F. Qian, S. Zhao, and Y. Zhang, "M-GWNN: multi-granularity graph wavelet neural networks for semi-supervised node classification," *Neurocomputing*, vol. 453, pp. 524–537, 2021.

[38] Y. Zhang, T. Cheng, and Y. Ren, "A graph deep learning method for short-term traffic forecasting on large road networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 10, pp. 877–896, 2019.

[39] A. Rafiq, T. A. Khan, M. Afaq, and W. C. Song, "Service function chaining and traffic steering in SDN using graph neural network," in *Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 500–505, IEEE, Jeju, Korea (South), June 2020.

[40] Y. Li, X. Fu, and Z. J. Zha, "Cross-patch graph convolutional network for image denoising," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4651–4660, Montreal, QC, Canada, October 2021.

[41] P. Sun, J. Lan, J. Li, Z. Guo, Y. Hu, and T. Hu, "Efficient flow migration for NFV with graph-aware deep reinforcement learning," *Computer Networks*, vol. 183, Article ID 107575, 2020.

[42] F. Habibi, M. Dolati, A. Khonsari, and M. Ghaderi, "Accelerating virtual network embedding with graph neural networks," in *Proceedings of the 2020 16th International Conference on Network and Service Management (CNSM)*, pp. 1–9, IEEE, Izmir, Turkey, November, 2020.

[43] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: a deep reinforcement learning approach with graph convolutional networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1040–1057, 2020.

[44] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," in *Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, IEEE, Waikoloa, HI, USA, December, 2019.

[45] J. Guo and C. Yang, "Learning power control for cellular systems with heterogeneous graph neural network," in *Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, Nanjing, China, March, 2021.

[46] K. Nakashima, S. Kamiya, K. Ohtsu, K. Yamamoto, T. Nishio, and M. Morikura, "Deep reinforcement learning-based channel allocation for wireless lans with graph convolutional networks," *IEEE Access*, vol. 8, pp. 31823–31834, 2020.

[47] K. Rusek, J. Suárez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet: leveraging graph neural networks for network modeling and optimization in SDN," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2260–2270, 2020.

[48] F. Geyer and G. Carle, "Learning and generating distributed routing protocols using graph-based deep learning," in *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, pp. 40–45, New York City, NY, USA, August, 2018.

[49] J. Zheng, Z. Zeng, and T. Feng, "GCN-ETA: high-efficiency encrypted malicious traffic detection," *Security and Communication Networks*, vol. 2022, Article ID 4274139, 2022.

[50] Z. Li, Y. Zhao, Y. Zhang, and Z. Zhang, "Multi-relational graph attention networks for knowledge graph completion," *Knowledge-Based Systems*, vol. 251, Article ID 109262, 2022.

[51] X. Tang, Y. Li, Y. Sun, H. Yao, P. Mitra, and S. Wang, "Transferring robustness for graph neural network against poisoning attacks," 2020, https://arxiv.org/abs/1908.07558.

[52] Y. Pan, X. Zhang, H. Jiang, and C. Li, "A network traffic classification method based on graph convolution and lstm," *IEEE Access*, vol. 9, pp. 158261–158272, 2021.

[53] X. Song, H. Li, W. Gao et al., "Augmented multicenter graph convolutional network for COVID-19 diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 6499–6509, 2021.

[54] C. Chen, W. Ma, M. Zhang et al., "Graph heterogeneous multi-relational recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, pp. 3958–3966, Chong Chen, China, May, 2021.

[55] Y. Zhu, J. Ma, C. Yuan, and X. Zhu, "Interpretable learning based dynamic graph convolutional networks for alzheimer's disease analysis," *Information Fusion*, vol. 77, pp. 53–61, 2022.

[56] H. Jiang, P. Cao, M. Xu, J. Yang, and O. Zaiane, "Hi-GCN: a hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction," *Computers in Biology and Medicine*, vol. 127, Article ID 104096, 2020.

[57] D. Arifoglu, H. N. Charif, and A. Bouchachia, "Detecting indicators of cognitive impairment via graph convolutional networks," *Engineering Applications of Artificial Intelligence*, vol. 89, Article ID 103401, 2020.

[58] C. Mao, L. Yao, and Y. Luo, "MedGCN: medication recommendation and lab test imputation via graph convolutional networks," *Journal of Biomedical Informatics*, vol. 127, Article ID 104000, 2022.

[59] J. Shi, R. Wang, Y. Zheng, Z. Jiang, H. Zhang, and L. Yu, "Cervical cell classification with graph convolutional network," *Computer Methods and Programs in Biomedicine*, vol. 198, Article ID 105807, 2021.

[60] H. Zhang, L. Chen, J. Cao, X. Zhang, and S. Kan, "A combined traffic flow forecasting model based on graph convolutional network and attention mechanism," *International Journal of Modern Physics C*, vol. 32, no. 12, Article ID 2150158, 2021.

[61] S. Yin, Q. Peng, H. Li et al., "Multi-instance deep learning with graph convolutional neural networks for diagnosis of kidney diseases using ultrasound imaging," in *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging and Clinical Image-Based Procedures*, pp. 146–154, Springer, Cham, Champa, 2019.

[62] Z. Chen, J. Liu, M. Zhu, P. Y. Woo, and Y. Yuan, "Instance importance-aware graph convolutional network for 3D medical diagnosis," *Medical Image Analysis*, vol. 2022, Article ID 102421, 21 pages, 2022.

[63] B. Yang, Y. Kang, L. Zhang, and H. Li, "GGAC: multi-relational image gated GCN with attention convolutional binary neural tree for identifying disease with chest X-rays," *Pattern Recognition*, vol. 120, Article ID 108113, 2021.

[64] Y. D. Zhang, S. C. Satapathy, D. S. Guttery, J. M. Górriz, and S. H. Wang, "Improved breast cancer classification through combining graph convolutional network and convolutional neural network," *Information Processing and Management*, vol. 58, no. 2, Article ID 102439, 2021.

[65] J. Wu, "Learning differential diagnosis of skin conditions with Co-occurrence supervision using graph convolutional

networks," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020. MICCAI 2020,* Vol. 12262, Springer, Cham, Champa, 2020.

[66] P. Xuan, H. Cui, H. Zhang et al., "Dynamic graph convolutional autoencoder with node-attribute-wise attention for kidney and tumor segmentation from CT volumes," *Knowledge-Based Systems*, vol. 236, Article ID 107360, 2022.

[67] F. Guo, Z. Li, Z. Xin, X. Zhu, L. Wang, and J. Zhang, "Dual graph U-nets for hyperspectral image classification," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 8160–8170, 2021.

[68] S. Shehnepoor, R. Togneri, W. Liu, and M. Bennamoun, "HIN-RNN: a graph representation learning neural network for fraudster group detection with no handcrafted features," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 2021, Article ID 3123876, 14 pages, 2021.

[69] T. Ma, L. Huang, Q. Lu, and S. Hu, "KR-GCN: knowledge-aware reasoning with graph convolution network for explainable recommendation," 2022, https://arxiv.org/abs/1905.04413.

[70] J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, and L. Cui, "Enhancing social recommendation with adversarial graph convolutional networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3727–3739, 2022.

[71] F. Liu, Z. Cheng, L. Zhu, C. Liu, and L. Nie, "A2-GCN: an attribute-aware attentive GCN model for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 9, pp. 4077–4088, 2022.

[72] Y. Xiao, C. Li, and V. Liu, "DFM-GCN: a multi-task learning recommendation based on a deep graph neural network," *Mathematics*, vol. 10, no. 5, p. 721, 2022.

[73] H. Tang, G. Zhao, X. Bu, and X. Qian, "Dynamic evolution of multi-graph based collaborative filtering for recommendation systems," *Knowledge-Based Systems*, vol. 228, Article ID 107251, 2021.

[74] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[75] Z. Zhang, Y. Liu, and F. Sun, "Fusion graph convolutional collaborative filtering," in *Pacific Rim International Conference on Artificial Intelligence*, pp. 574–584, Springer, Cham, Champa, 2021.

[76] C. Yang, A. Pal, A. Zhai et al., "Multisage: empowering gcn with contextualized multi-embeddings on web-scale multipartite networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2434–2443, New York City, NY, USA, August, 2020.

[77] H. Wang, F. Zhang, J. Wang et al., "Ripplenet: propagating user preferences on the knowledge graph for recommender systems," in *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 417–426, New York City, NY, USA, October, 2018.

[78] J. Liao, W. Zhou, F. Luo et al., "SocialLGN: light graph convolution network for social recommendation," *Information Sciences*, vol. 589, pp. 595–607, 2022.

[79] L. Guo, L. Tang, T. Chen, L. Zhu, Q. V. H. Nguyen, and H. Yin, "DA-GCN: a domain-aware attentive graph convolution network for shared-accountcross-domain sequential recommendation," 2021, https://arxiv.org/abs/2105.03300.

[80] H. Liu, Y. Wei, J. Yin, and L. Nie, "HS-GCN: hamming spatial graph convolutional networks for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 2022, Article ID 3158317, 1 page, 2022.

[81] Y. Guo, X. Yin, X. Zhao, D. Yang, and Y. Bai, "Hyperspectral image classification with SVM and guided filter," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 56–59, 2019.

[82] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.

[83] Y. Luo, J. Zou, C. Yao, X. Zhao, T. Li, and G. Bai, "HSI-CNN: a novel convolution neural network for hyperspectral image," in *Proceedings of the 2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 464–469, IEEE, Shanghai, China, July, 2018.

[84] L. Mou, X. Lu, X. Li, and X. X. Zhu, "Nonlocal graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 12, pp. 8246–8257, 2020.

[85] J. Y. Yang, H. C. Li, W. S. Hu, L. Pan, and Q. Du, "Adaptive cross-attention-drivenspatial-spectral graph convolutional network for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[86] J. Qu, Y. Xu, W. Dong, Y. Li, and Q. Du, "Dual-branch difference amplification graph convolutional network for hyperspectral image change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[87] A. Sha, B. Wang, X. Wu, and L. Zhang, "Semisupervised classification for hyperspectral images using graph attention networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 1, pp. 157–161, 2021.

[88] Y. Ding, X. Zhao, Z. Zhang, W. Cai, and N. Yang, "Multiscale graph sample and aggregate network with context-aware learning for hyperspectral image classification," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4561–4572, 2021b.

[89] K. Xu, Y. Zhao, L. Zhang, C. Gao, and H. Huang, "Spectral–spatial residual graph attention network for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[90] S. Saha, L. Mou, X. X. Zhu, F. Bovolo, and L. Bruzzone, "Semisupervised change detection using graph convolutional network," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 4, pp. 607–611, 2021.

[91] A. Ismail and M. Awad, "BLDNet: a semi-supervised change detection building damage framework using graph convolutional networks and urban domain knowledge," 2022, https://arxiv.org/abs/2201.10389.

[92] S. Li, L. Zhang, H. Feng et al., "MutagenPred-GCNNs: a graph convolutional neural network-based classification model for mutagenicity prediction with data-driven molecular fingerprints," *Interdisciplinary Sciences: Computational Life Sciences*, vol. 13, no. 1, pp. 25–33, 2021b.

[93] Y. Shen, H. Fu, Z. Du et al., "GCN-denoiser: mesh denoising with graph convolutional networks," *ACM Transactions on Graphics*, vol. 41, no. 1, pp. 1–14, 2022.

[94] S. Ouyang and Y. Li, "Combining deep semantic segmentation network and graph convolutional neural network for semantic segmentation of remote sensing imagery," *Remote Sensing*, vol. 13, no. 1, p. 119, 2020.

[95] N. Kim, B. Kang, and Y. Cho, "Split GCN: effective interactive annotation for segmentation of disconnected instance," 2021, https://arxiv.org/abs/2112.06454.

[96] D. Zhan, D. Liang, H. Jin, and X. Wu, "MBBOS-GCN: minimum bounding box over-segmentation—graph convolution 3D point cloud deep learning model," *Journal of Applied Remote Sensing*, vol. 16, no. 1, Article ID 16502, 2022.

[97] K. Wang, S. C. Han, S. Long, and J. Poon, "ME-GCN: multi-dimensionaledge-embedded graph convolutional networks for semi-supervised text classification," 2022, https://arxiv.org/abs/2204.04618#:%7E:text=10%20Apr%202022%5D-,ME%2DGCN%3A%20Multi%2Ddimensional%20Edge%2DEmbedded%20Graph%20Convolutional,for%20Semi%2Dsupervised%20Text%20Classification%26text=Abstract%3A%20Compared%20to%20sequential%20learning,for%20semi%2Dsuper.

[98] X. Wei, R. Yu, and J. Sun, "View-gcn: view-based graph convolutional network for 3d shape analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1850–1859, Seattle, WA, USA, June 2020.

[99] X. Yao, F. Xu, M. Gu, and P. Wang, "M-GCN: brain-inspired memory graph convolutional network for multi-label image recognition," *Neural Computing and Applications*, vol. 34, no. 8, pp. 6489–6502, 2022.

[100] M. Nakao, M. Nakamura, and T. Matsuda, "Image-to-Graph convolutional network for 2D/3D deformable model registration of low-contrast organs," *IEEE Transactions on Medical Imaging*, vol. 41, no. 12, pp. 3747–3761, 2022.

[101] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," 2017, https://arxiv.org/abs/1707.01926.

[102] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020.

[103] J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhao, and H. Li, "AST-GCN: attribute-augmented spatiotemporal graph convolutional network for traffic forecasting," *IEEE Access*, vol. 9, pp. 35973–35983, 2021.

[104] H. Zhu, Y. Xie, W. He et al., "A novel traffic flow forecasting method based on RNN-GCN and BRB," *Journal of Advanced Transportation*, vol. 2020, Article ID 7586154, 11 pages, 2020.

[105] Y. Hu, H. Shen, W. Liu, F. Min, X. Qiao, and K. Jin, "A graph convolutional network with multiple dependency representations for relation extraction," *IEEE Access*, vol. 9, pp. 81575–81587, 2021.

[106] K. Sun, Y. Li, H. Zhang, C. Guo, L. Yuan, and Q. Hu, "Syntax–Aware graph convolutional network for the recognition of Chinese implicit inter-sentence relations," *The Journal of Supercomputing*, vol. 78, no. 14, pp. 16529–16552, 2022.

[107] J. Zhou, J. X. Huang, Q. V. Hu, and L. He, "Sk-gcn: modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification," *Knowledge-Based Systems*, vol. 205, Article ID 106292, 2020.

[108] M. Harl, S. Weinzierl, M. Stierle, and M. Matzner, "Explainable predictive business process monitoring using gated graph neural networks," *Journal of Decision Systems*, vol. 29, no. 1, pp. 312–327, 2020.

[109] L. Song, Z. Wang, M. Yu, Y. Zhang, R. Florian, and D. Gildea, "Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks," 2018, https://arxiv.org/abs/1809.02040.

[110] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," 2017, https://arxiv.org/abs/1703.04826.

[111] Z. Ye, G. Jiang, Y. Liu, Z. Li, and J. Yuan, "Document and word representations generated by graph convolutional network and bert for short text classification," in *ECAI 2020*, pp. 2275–2281, IOS Press, Amsterdam, Netherlands, 2020.

[112] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[113] F. D. Malliaros and K. Skianis, "Graph-based term weighting for text categorization," in *Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining 2015*, pp. 1473–1479, New York City, NY, USA, August, 2015.

[114] A. Pal, M. Selvakumar, and M. Sankarasubbu, "Multi-label text classification using attention-based graph neural network," 2020, https://arxiv.org/abs/2003.11644.

[115] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 1, pp. 7370–7377, Laguna Hills, CA, USA, July, 2019.

[116] Q. Wang, K. Zhang, and M. A. Asghar, "Skeleton-based ST-GCN for human action recognition with extended skeleton graph and partitioning strategy," *IEEE Access*, vol. 10, pp. 41403–41410, 2022.

[117] Y. Yuan and Z. Bar-Joseph, "GCNG: graph convolutional networks for inferring gene interaction from spatial transcriptomics data," *Genome Biology*, vol. 21, no. 1, pp. 300–316, 2020.

[118] A. Rao, S. Vg, T. Joseph, S. Kotte, N. Sivadasan, and R. Srinivasan, "Phenotype-driven gene prioritization for rare diseases using graph convolution on heterogeneous networks," *BMC Medical Genomics*, vol. 11, no. 1, pp. 57–12, 2018.

[119] S. Harada, H. Akita, M. Tsubaki et al., "Dual graph convolutional neural network for predicting chemical networks," *BMC Bioinformatics*, vol. 21, no. 3, pp. 94–13, 2020.

[120] J. Zhang, B. Liu, Z. Wang, K. Lehnert, and M. Gahegan, "DeepPN: a deep parallel neural network based on convolutional neural network and graph convolutional network for predicting RNA-protein binding sites," *BMC Bioinformatics*, vol. 23, no. 1, p. 257, 2022.

[121] N. Zaki, H. Singh, and E. A. Mohamed, "Identifying protein complexes in protein-protein interaction data using graph convolutional network," *IEEE Access*, vol. 9, pp. 123717–123726, 2021.

[122] C. Peng, S. Dieck, A. Schmid et al., "CADA: phenotype-driven gene prioritization based on a case-enriched knowledge graph," *NAR Genomics and Bioinformatics*, vol. 3, no. 3, p. 78, 2021.

[123] L. Liu, H. Mamitsuka, and S. Zhu, "HPOFiller: identifying missing protein–phenotype associations by graph convolutional network," *Bioinformatics*, vol. 37, no. 19, pp. 3328–3336, 2021.

[124] J. B. Lee, R. Rossi, and X. Kong, "Graph classification using structural attention," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1666–1674, New York City, NY, USA, July, 2018.

[125] F. P. Such, S. Sah, M. A. Dominguez et al., "Robust spatial filtering with graph convolutional neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 884–896, 2017.

[126] N. De Cao and T. Kipf, "MolGAN: an implicit generative model for small molecular graphs," 2018, https://arxiv.org/abs/1805.11973.

[127] G. Xiaoai, X. Yujing, L. Lin, and T. Lin, "An overview of disease prediction based on graph convolutional neural network," in *Proceedings of the 2021 6th International Conference on Intelligent Information Processing*, pp. 27–32, New York City, NY, USA, July, 2021.

[128] W. Torng and R. B. Altman, "Graph convolutional neural networks for predicting drug-target interactions," *Journal of Chemical Information and Modeling*, vol. 59, no. 10, pp. 4131–4149, 2019.

[129] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the International Conference on Machine Learning*, pp. 1263–1272, PMLR, New York City, NY, USA, July, 2017.

[130] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, 2018.

[131] Z. Xiao and Y. Deng, "Graph embedding-based novel protein interaction prediction via higher-order graph convolutional network," *PLoS One*, vol. 15, no. 9, Article ID 238915, 2020.

[132] M. Réau, N. Renaud, L. C. Xue, and A. M. J. J. Bonvin, "DeepRank-GNN: a graph neural network framework to learn patterns in protein–protein interfaces," *Bioinformatics*, vol. 39, no. 1, p. 759, 2023.

[133] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec, "Graph convolutional policy network for goal-directed molecular graph generation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[134] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1416–1424, New York City, NY, USA, July, 2018.

[135] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proceedings of the International Conference on Machine Learning*, pp. 1725–1735, PMLR, New York City, NY, USA, November, 2020.

[136] Q. Li, Z. Han, and X. M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the Thirty-Second AAAI conference on artificial intelligence*, Laguna Hills, CA, USA, April, 2018.

[137] Y. You, T. Chen, Z. Wang, and Y. Shen, "When does self-supervision help graph convolutional networks?" in *Proceedings of the international conference on machine learning*, pp. 10871–10880, PMLR, New York City, NY, USA, November, 2020.

[138] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: graph convolutional neural networks with complex rational spectral filters," *IEEE Transactions on Signal Processing*, vol. 67, no. 1, pp. 97–109, 2019.

[139] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, "Disentangled graph convolutional networks," in *Proceedings of the International conference on machine learning*, pp. 4212–4221, PMLR, New York City, NY, USA, May, 2019.

[140] Y. Yang, J. Qiu, M. Song, D. Tao, and X. Wang, "Distilling knowledge from graph convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7074–7083, Seattle, WA, USA, June 2020.

[141] Z. Chen, B. Zhao, Y. Wang, Z. Duan, and X. Zhao, "Multitask learning and GCN-based taxi demand prediction for a traffic road network," *Sensors*, vol. 20, no. 13, Article ID 20133776, 2020.

[142] J. Liu, G. P. Ong, and X. Chen, "GraphSAGE-based traffic speed forecasting for segment network with sparse data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1755–1766, 2022.

[143] H. P. Maretic, M. El Gheche, and P. Frossard, "Graph heat mixture model learning," in *Proceedings of the 2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 1003–1007, IEEE, Pacific Grove, CA, USA, October, 2018.

[144] A. Micheli, "Neural network for graphs: a contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009.