Artificial Intelligence Review
https://doi.org/10.1007/s10462-023-10577-2

Attention-based graph neural networks: a survey

Chengcheng Sun[1] · Chenhao Li[1] · Xiang Lin[1] · Tianji Zheng[1] · Fanrong Meng[1,2] · Xiaobin Rui[1] · Zhixiao Wang[1,2]

Abstract

Graph neural networks (GNNs) aim to learn well-trained representations in a lower-dimension space for downstream tasks while preserving the topological structures. In recent years, attention mechanism, which is brilliant in the fields of natural language processing and computer vision, is introduced to GNNs to adaptively select the discriminative features and automatically filter the noisy information. To the best of our knowledge, due to the fast-paced advances in this domain, a systematic overview of attention-based GNNs is still missing. To fill this gap, this paper aims to provide a comprehensive survey on recent advances in attention-based GNNs. Firstly, we propose a novel two-level taxonomy for attention-based GNNs from the perspective of development history and architectural perspectives. Specifically, the upper level reveals the three developmental stages of attention-based GNNs, including graph recurrent attention networks, graph attention networks, and graph transformers. The lower level focuses on various typical architectures of each stage. Secondly, we review these attention-based methods following the proposed taxonomy in detail and summarize the advantages and disadvantages of various models. A model characteristics table is also provided for a more comprehensive comparison. Thirdly, we share our thoughts on some open issues and future directions of attention-based GNNs. We hope this survey will provide researchers with an up-to-date reference regarding applications of attention-based GNNs. In addition, to cope with the rapid development in this field, we intend to share the relevant latest papers as an open resource at https://github.com/sunxiaobei/awesome-attention-based-gnns.

Keywords Graph neural networks · Attention mechanism · Graph attention networks · Graph transformers · Graph representation learning

Chengcheng Sun and Chenhao Li have contributed equally to this work.

✉ Zhixiao Wang
zhxwang@cumt.edu.cn

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, Jiangsu, China

[2] Mine Digitization Engineering Research Center, Ministry of Education of the People's Republic of China, Xuzhou 221116, Jiangsu, China

Published online: 21 August 2023

1 Introduction

As a typical non-Euclidean data structure, graphs are ubiquitous in the real world, such as social networks, transportation networks, citation networks, and protein interaction networks. Recent deep learning research on graphs has attracted more and more attention due to the rich expressive power of deep learning models (Zhou et al. 2020). Graph neural networks (GNNs (Scarselli et al. 2008; Bruna et al. 2014)), as a kind of deep graph representation learning method, aim to learn nodes/edges/graphs-level representations in a lower-dimension space for various downstream tasks (e.g., node classification (Welling and Kipf 2016), link prediction (Lv et al. 2021), community detection (Su et al. 2022), and graph classification (Ying et al. 2021)). Different from the traditional graph embedding methods (i.e., matrix factorization (Roweis and Saul 2000) and random walk-based methods (Perozzi et al. 2014; Tang et al. 2015; Grover and Leskovec 2016)), GNNs learn both the structural and attribute features of a graph through different neural networks. Due to this inherent advantage, GNNs can naturally be used to deal with graph-structured data with a large amount of attribute information (Yang et al. 2021c).

Existing GNNs (Wu et al. 2020b), originating from graph convolution (Welling and Kipf 2016), graph diffusion (Klicpera et al. 2019), graph attention (Veličković et al. 2018) or other various mechanisms, are usually divided into two classical categories: spectral domain (Bruna et al. 2014; Defferrard et al. 2016; Welling and Kipf 2016) and spatial domain (Hamilton et al. 2017; Veličković et al. 2018; Xu et al. 2018a). Spectral graph neural networks realize topological convolution operations based on spectral graph theory and graph signal processing. Typical methods include SpectralCNN (Bruna et al. 2014), ChebNet (Defferrard et al. 2016), and GCN (Welling and Kipf 2016). Spectral GNN is based on the Fourier transform and strongly depends on the graph Laplacian matrix (Welling and Kipf 2016). Once the graph structure is determined, it is difficult to extend such methods to another new graph. Spatial graph neural networks directly aggregate and update the information of nodes from neighborhoods iteratively under message-passing neural networks (MPNN (Gilmer et al. 2017)). Considering the attributes of nodes/edges/graphs, Graph Networks (GraphNet (Battaglia et al. 2018)) generalizes and extends various graph neural networks under message-passing mechanism and supports constructing complex architectures from simple building blocks. Most GNNs treat neighborhood nodes equally without considering the contributions of different nodes when aggregating and propagating information. However, real-world graphs are often noisy with connections between unrelated nodes, which causes GNNs to learn suboptimal representations without distinguishing nodes in neighborhoods (Kim and Oh 2021).

To avoid noisy signals, GAT (Veličković et al. 2018), a classical spatial graph neural network, introduces an attention mechanism into the graph neural network to learn the importance of different neighbors. Thereafter, attention becomes a popular mechanism used in a wide range of graph deep neural network architectures for better representation (Kim and Oh 2021). Till now, there are a large number of attention-based graph neural networks. Attention in GNNs can be applied to different node neighborhoods, different parts of structures, and different representations of nodes, edges, and graphs.

In 2019, a survey reviews several attention models in graphs from the perspective of graph types, attention types, and task types (Lee et al. 2019b). After that, reviews of attention models in graphs have been vacant. Most reviews focus more on the taxonomy and application of existing GNNs (Wu et al. 2020b; Zhou et al. 2022; Georgousis et al. 2021; Wu et al. 2021b; Gao et al. 2022; Jiang and Luo 2021; Cini et al. 2022), and they also describe GNNs in both the spectral domain and spatial domain. A comprehensive survey (Wu et al. 2020b) divides the state-of-the-art GNNs into four categories, including recurrent graph neural networks, convolutional graph neural networks, graph autoencoders, and spatial-temporal graph neural networks. To better address the challenges that exist in GNNs, a survey discusses several fundamental challenge problems from both theoretical and practical perspectives (Georgousis et al. 2021). Other reviews cover more focused aspects of specific application domains, such as GNN4NLP (Wu et al. 2021b), GNN4RS (Gao et al. 2022), GNN4Traffic (Jiang and Luo 2021), and GNN4IoT (Cini et al. 2022). Recently, an overview provides a review of various transformers for graphs from an architectural perspective (Min et al. 2022). Several attention-based GNNs are usually covered in the above reviews on GNNs, but they only list a few of the core literature on attention-based GNNs without further classification. By contrast, we aim to provide a systematic overview of attention-based GNNs from development history and architectural perspectives in this survey. Considering the rapid evolution of this field, it’s imperative to sort out the existing attention-based GNNs and systematically analyze advanced methods for both academics and practitioners. However, to the best of our knowledge, an up-to-date, and comprehensive overview of attention-based GNNs is missing. To fill this gap, we present a novel two-level taxonomy from both developmental stages and architectural perspectives to systematically review attention-based GNNs.

Existing attention-based graph neural networks can be divided into three stages, including Graph Recurrent Attention Networks (GRANs (Li et al. 2016; Zhang et al. 2018; Liao et al. 2019)), Graph Attention Networks (GATs (Veličković et al. 2018; Kim and Oh 2021)), and Graph Transformers (Nguyen et al. 2019; Zhang et al. 2020a), as shown in Fig. 1. GRANs introduce the attention mechanism and RNN into graph neural networks. Due to the inherent limitations of RNN, the calculation of each step of GRANs usually depends on the results of the previous step. On the contrary, GATs do not need to rely on the previous results and can implement parallel operations. Most GATs focus on nodes in local neighborhoods with local attention, which distinguish the importance of different neighbors. However, GRANs and GATs also suffer from a common limitation: they cannot capture remote dependencies. Consider capturing remote messages, Graph Transformers can learn higher-order graph information directly with global attention. Each stage has its representative and typical methods. Therefore, we further categorize these methods from an architectural perspective for each stage.

In GRANs, most methods are inspired by attention-based Recurrent Neural Networks (RNNs). Therefore, we naturally divide these methods into two subclasses. One is based on Gated Recurrent Units (GRU) (Zhang et al. 2018), and the other is attention-based Long Short-Term Memory (LSTM) (Xu et al. 2018b). However, both of those methods are suffered from the long-time dependency bottleneck and the order problem that exists in RNNs.

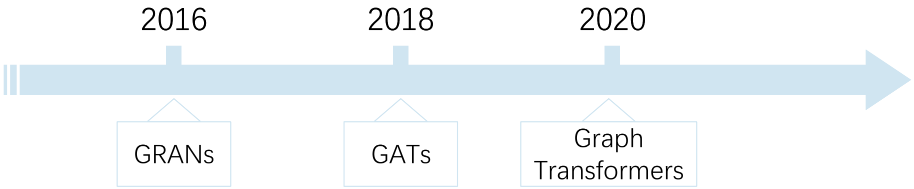


Fig. 1 Three developmental stages of attention-based graph neural networks

GATs assign different weights to nodes in the feature aggregation steps according to their influences. GAT (Veličković et al. 2018) is the pioneer of this stage. Later, a great variety of GATs, including C-GAT (Wang et al. 2019a), GATv2 (Brody et al. 2021), and SuperGAT (Kim and Oh 2021), adopt different attention strategies to GNNs. We distinguish GATs from the perspective of whether the attention mechanism is in the neural network layer, namely intra-layer GATs and inter-layer GATs. For intra-layer GATs, the attention function is used to calculate the weights of different nodes in local neighborhoods, and then dynamically update the representation of nodes. For inter-layer GATs, attention is usually regarded as an operation of feature combination, which is used to select features from different levels, different channels, different views, or different time slices.

Graph Transformers (Rong et al. 2020) can learn higher-order graph properties directly, different from previous methods with local attention. In the past two years, Graph Transformers have developed rapidly in the field of graph deep learning, especially in the task of graph classification. However, the globally-connected self-attention mechanism in graph transformers makes it necessary to update the weights of the whole network continuously in the process of end-to-end model training (Ying et al. 2021; Kim et al. 2021b). This prevents them from exploiting sparsity in the graph topology, leading to excessively high computational complexity.

In a word, this paper aims to provide a systematic and comprehensive review of contemporary attention-based graph neural networks. Our contributions can be summarized as follows.

- This paper proposes a novel two-level taxonomy for attention-based GNNs from the perspective of development history and architecture. Specifically, the upper level reveals the three developmental stages of attention-based GNNs, including GRANs, GATs, and Graph Transformers. The lower level focuses on various typical architectures of each stage.
- This paper comprehensively and systematically summarizes the latest works on attention-based GNNs, which makes up for the lack of literature in this hot direction. For each subcategory, we provide a detailed introduction and an in-depth comparison to reveal the advantages and disadvantages of various models.
- This paper also provides open issues and challenges of attention-based GNNs as insights for future research directions to advance this field, which will provide researchers with an up-to-date reference about attention-based GNNs.

The rest of this survey paper is organized as follows. Section 2 gives the preliminaries and notations of attention-based GNNs. Section 3 provides a two-level taxonomy from the history and architectural perspectives. Section 4 presents a technical overview of GRANs. We introduce the GATs in both Sect. 5 and Sect. 6. Section 7 overviews Graph Transformers. In Sect. 8, we summarize and compare the characteristics of different models in subclasses. We suggest promising future research directions in Sect. 9 and Sect. 10 comes to the conclusion of this paper.

2 Preliminaries and notations

2.1 Graph

A graph is a data structure defining a set of nodes and their relationships. As shown in Fig. 2, unlike sequence-structured data in NLP and grid-structured data in CV, graph-structured data is extremely complex. Especially, the number of neighbor nodes for each node is irregular. With the growth of the graph scale, the number of nodes in the graph often increases exponentially.

Let $G = \langle V, E \rangle$ represents an attributed graph with a feature matrix as X where $V = \left\{ v_1, v_2, \cdots, v_n \right\}$ refers to the set of nodes, $E = \left\{ e_1, e_2, \cdots, e_m \right\}$ refers to the set of edges between nodes. We denote $n = |V|$ as the number of nodes and $m = |E|$ as the number of edges. $X \in \mathbb{R}^{n \times dim}$ represents the feature matrix and its row element $\mathbf{x}_i \in \mathbb{R}^{dim}$ represents the feature vector of node v_i with the dimension dim. The adjacency matrix could represent the connection relationship between nodes, defined as $A \in \{0, 1\}^{n \times n}$, where $A_{ij} = 1$ means that there is an edge e_{ij} between node v_i and v_j. We use $d_i = \sum_{j \in \Gamma_i} A_{ij}$ to denote the degree of node v_i, where Γ_i is the neighborhood nodes of node v_i. For a multi-relational graph, we extend the edge notation with edge type $r \in R$, as e_{ij}^{r}. Correspondingly, edge and graph can have attributes as well, defined as X_e, X_g. If not specified, for convenience we default X to represent the node attribute matrix.

2.2 Graph neural networks

Graph Neural Networks, a series of methods in Graph Representation Learning (GRL) based on deep learning, take a source graph G as input, with an adjacency matrix A and feature matrix X. GNNs aim to learn a potential representation embedding vectors $Z \in \mathbb{R}^{n \times dim'}$ in low dimensionality, i.e., $dim' \ll dim$, which will play a key role in downstream tasks, such as node classification, link prediction, and graph classification. See Fig. 3 for an illustration.

From a spatial perspective, MPNN (Gilmer et al. 2017) can be seen as a more constrained instance of GraphNet (Battaglia et al. 2018), considering only the attributes of the nodes. They propagate information over the graph by a local diffusion process (Stachenfeld et al. 2020). The GraphNet block contains three update functions and three aggregation functions, while MPNN has only one pair of such functions.

Taking node features X and graph structure A as inputs, GNNs can be defined as $Z^{out} = \mathcal{F}(X, \ A, \ \Theta)$, where Θ refers to the learnable weight parameter and $\mathcal{F}(\cdot)$ denotes the GNN encoders such as GCNs, GATs even MLPs. Mathematically, we can define the general framework of graph neural networks with the message-passing mechanism as follows:

$$M_{v_i}^{l} = Agg^{l}\left(\left\{ H_{v_j}^{l-1}, \forall v_j \in \Gamma_i \right\} \right) \tag{1}$$

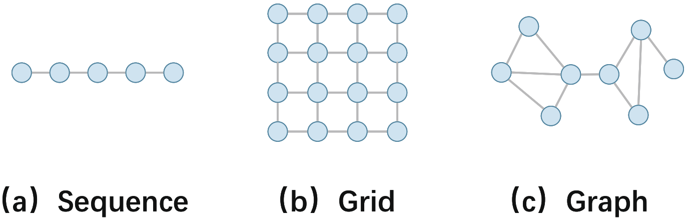


Fig. 2 Examples of sequence-structured, grid-structured, and graph-structured data

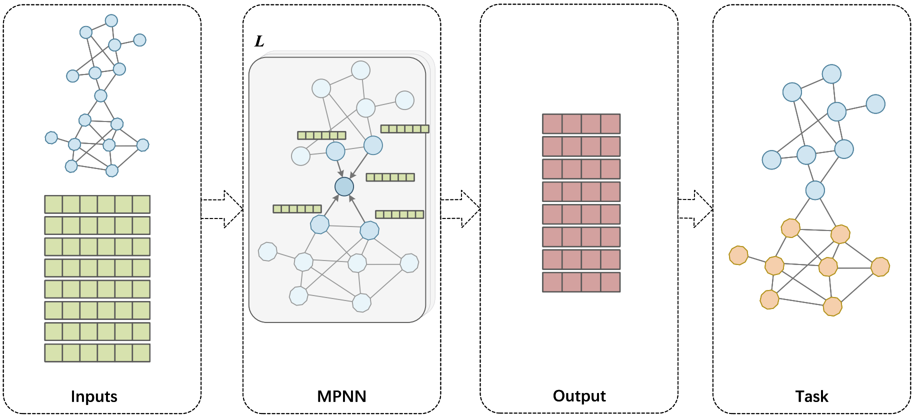


Fig. 3 The architecture of GNNs with the message-passing mechanism

$$H_{v_i}^{l} = Comb^{l}\left(H_{v_i}^{l-1},\ M_{v_i}^{l}\right) \tag{2}$$

where $H^l \in \mathbb{R}^{n\times dim}$ represents the node features in the l^{th} layer starting from the initial node features $H^0 = X$. Γ_i denotes the neighborhood node v_i with a set of neighbors. $M_{v_i}^{l}$ refers to the received messages from neighbors in the neighborhood Γ_i of the current node v_i in the l^{th} layer. From the view of the message-passing mechanism, GNN can be defined as two important functions: $Agg(\cdot)$ and $Comb(\cdot)$. $Agg(\cdot)$ is the aggregator function to aggregate the messages from the neighbors of each node in the graph, while $Comb(\cdot)$ is the combined (or update) function to update the node representations by combining the received messages from neighbors and the representations of the current node in the former layer.

For the node classification task in Fig. 3, we can take $Z^{out} = H^L$ as the output of graph neural networks after L layers. For graph-level tasks, an additional readout function $Read(\cdot)$ should be defined to read out the final representation H_G of the entire graph or subgraph. There are a series of readout functions, such as average, sum, and max functions, which can be defined as below:

$$H_G = Read(\{H_v^l,\ \forall\, v \in V\}) \tag{3}$$

Take GCN for example, which is the most popular graph neural network due to its ease of understanding from the spatial domain as well as the theoretical basis from the spectral domain (Xia et al. 2021a). GCN iteratively aggregates and updates the representations of nodes in the graph through a propagation rule (Welling and Kipf 2016) defined as:

$$H^l = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{l-1}\Theta^l\right) \tag{4}$$

where $\widetilde{A} = A + I_n$ denotes the adjacency matrix with self-loop and $\widetilde{D}$ represents the corresponding diagonal degree matrix. I_n refers to the identity matrix, and $\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$ is a renormalization trick inspired by the symmetric normalized graph Laplacian $L^{sn} = I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$. Θ is a trainable weight parameter and $\sigma(\cdot)$ is a non-linear activation function $ReLU(x) = \max(0, x)$. For the message-passing mechanism in GCN, we rewrite the above formula from the view of nodes as follows:

$$H_{v_i}^{l} = \sigma\left(\sum_{j\in\Gamma_i} \frac{\widetilde{A}_{ij}}{\sqrt{d_i d_j}} H_{v_j}^{l-1}\Theta^{l} + \frac{\widetilde{A}_{ii}}{\sqrt{d_i d_i}} H_{v_i}^{l-1}\Theta^{l}\right) \tag{5}$$

where d_i refers to the degree of node v_i. *Agg* function in GCN is defined as the average of the neighbor node representations with a normalization constant based on degree. *Comb* function is defined as a simple summation function.

2.3 Attention Mechanisms

Attention is a mechanism based on the recognition process of the human visual system, which imitates the human cognitive awareness about specific information to focus more on the critical aspects of data (Guo et al. 2022; Brauwers and Frasincar 2021). The attention mechanisms have been successfully applied to various application fields in CV, NLP, and GRL. Inspired by attention in CV (Guo et al. 2022) and NLP (Chaudhari et al. 2021), we give a generalized definition of attention and then decompose the attention process into three effective functions, i.e., alignment function, distribution function, and weighted sum function.

The attention mechanism with different attention functions can be defined under a generalized framework:

$$Attention = f(g(X),\ X) \tag{6}$$

where $g(\cdot)$ refers to an attention function to generate attention which corresponds to capturing the important regions in the visual scene (Guo et al. 2022). $f(\cdot)$ means processing input data X to obtain important information based on the attention function $g(\cdot)$.

In more detail, the attention mechanism (shown in Fig. 4a) can be seen as a mapping of a sequence of keys K to an attention distribution α according to queries Q, and the keys have one-to-one corresponding values V (Chaudhari et al. 2021). In Fig. 4b, the attention function $g(\cdot)$ consists of two key components including an alignment function and a distribution function, while the $f(\cdot)$ is used as a weighted sum function to calculate the final attention value.

The alignment function is a process of calculating the attention alignment score, which is the core of the attention mechanism (Chaudhari et al. 2021). There are many classical alignment functions in Table 1, defined as:

$$scores = Sim(Q, K) \tag{7}$$

The distribution function converts the attention score to the attention coefficients α. We define the distribution function in a unified form as below:

$$\alpha = Norm(scores) \tag{8}$$

where, $Norm(\cdot)$ refers to a distribution function. The softmax function $softmax(\cdot)$ is the most commonly used distribution function, defined as:

$$softmax\left(x_i\right) = \frac{exp(x_i)}{\sum_{n=1}^{N} exp(x_n)} \tag{9}$$

Table 1 Summary of Common Alignment Functions

Function	Equation	Description
Cos similarity	$Cos(K, Q)$	$Cos(\cdot)$: Cosine similarity
Dot product	$Q^T K$	T: Matrix transpose
Scaled dot product	$\frac{Q^T K}{\sqrt{dim_K}}$	dim_K: Dimension of K
General	$Q^T \Theta K$	Θ: Trainable parameters
Biased general	$Q^T \Theta K + b$	b: Biase
Additive	$\omega^T\ Act(\Theta_1 Q + \Theta_2 K + b)$	ω: Trainable parameters
Concat	$\omega^T\ Act(\Theta[Q : K] + b)$	Act: Activate function

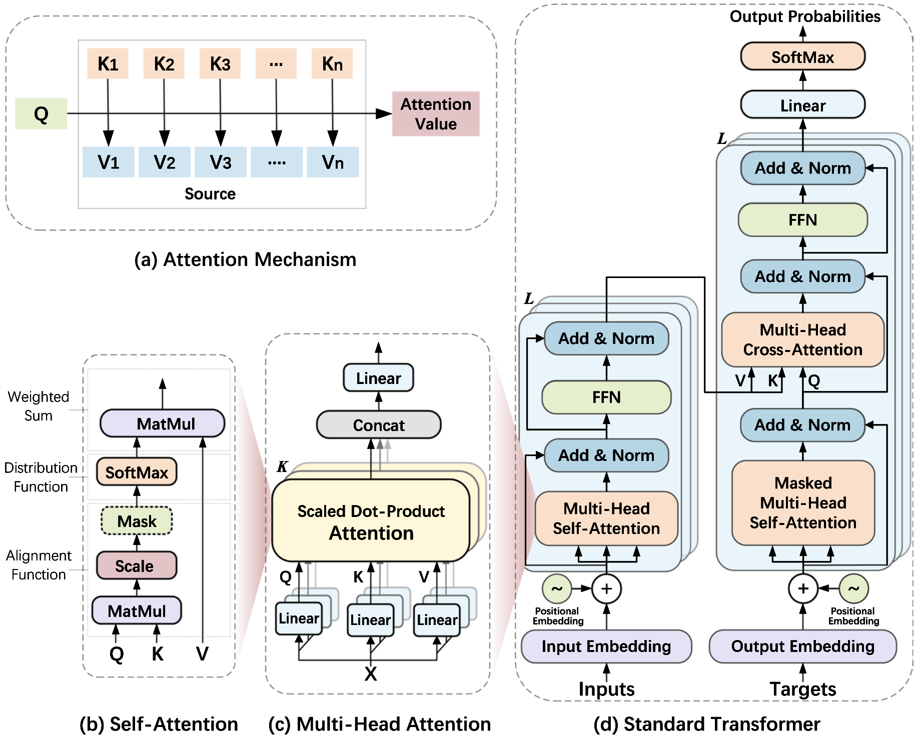


Fig. 4 The architecture of attention, self-attention, multi-head attention, and standard transformer

where, $x_i \in X$ represents the i-th element in X and N is the total number of elements in X. $exp(\cdot)$ is an exponential function. With a softmax function, the calculated attention score is normalized to the probability distribution in $[0, 1]$ and sum to 1.

Finally, an aggregation process for the attended representations with a weighted sum function to get the attention value:

$$Z = f(X) = \alpha\ V \tag{10}$$

Almost all existing attention mechanisms can be written into the above formulations under the generalized framework.

When an attention mechanism is used to compute a representation of a single sequence, it is commonly referred to as self-attention or intra-attention (Tay et al. 2020). Generally, the complete self-attention layer (SAL) with scaled dot product can be expressed as:

$$Z = SAL(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{dim_Q}}\right)V \tag{11}$$

The self-attention mechanism calculates the scaled dot product of the query with all the keys to find the similarity scores between them, and this product is normalized by the softmax function for obtaining the attention weight. The weighted sum function is used to get the attention value Z as outputs. Finally, the whole process of self-attention is shown in Fig. 4b. Further, we rewrite the self-attention layer under the generalized framework:

$$Q, K, V = Linear(X) \tag{12}$$

$$scores = \frac{QK^T}{\sqrt{dim_Q}} \tag{13}$$

$$\alpha = softmax(scores) \tag{14}$$

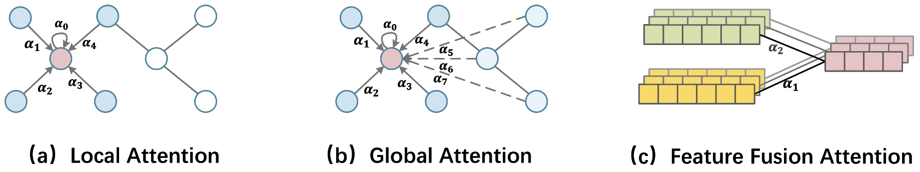
$$Z = \alpha\ V \tag{15}$$

where $Linear(\cdot)$ refers to a linear transformation, Q^T is a matrix transpose operation. $Q^T K$ is the dot product operation, which is widely used in self-attention to calculate the attention score.

The attention mechanism in GNNs allows the neural networks to learn a dynamic and adaptive aggregation of the neighborhood as well as enables the model to avoid or ignore noisy parts of the graph (Thekumparampil et al. 2018). Based on attention functions, we define three types of attention mechanisms in GNNs including local attention, global attention, and feature fusion attention.

Local attention focuses on local neighbors, that is, directly or locally connected neighbors. GAT (Veličković et al. 2018) learns different aggregating weights for each neighbor representation through local attention, as shown in Fig. 5a. Treating normalized attention coefficients as the relative weights between node pairs, attention-based GNNs aggregate and update the representations of nodes in the graph with a weighted sum function, then propagate them to higher layers (Klicpera et al. 2019). As shown in Fig. 5b, the self-attention mechanism in graph transformers can be viewed as passing messages among all nodes in the entire graph with global attention, regardless of the input graph connectivity (Chen et al. 2019). Different from local attention and global attention, feature fusion attention directly adopts the traditional attention mechanisms, focusing on feature fusion beyond the node, as shown in Fig. 5(c).

From the node view, let $h_i^l, h_j^l \in H^l$ refers to the representation of the node v_i and v_j in the $l^{th} \in L$ layer, and $h_i^0 = x_i^0$ as the input features of node v_i. As an initial step, a shared linear transformation with a weight matrix Θ^l is applied to every node in the graph. After the linear transformation, the alignment function obtains the attention alignment score, which



(a) Local Attention

(b) Global Attention

(c) Feature Fusion Attention

Fig. 5 Different attention mechanisms in GNNs. Local attention focuses only on the direct neighbors of the central node, while global attention focuses on all nodes in the graph. Feature fusion attention is a weighted approach from the perspective of feature fusion using an attention mechanism

indicates the importance of its local neighbors. In addition, the distribution function converts the attention score to the attention coefficient, which makes the attention coefficients comparable across different nodes. Finally, the weighted sum function is used to update and aggregate the representations of nodes in the graph. The above local attention layer can be defined as follows:

$$h_i^l = \Theta^l x_i^l \; h_j^l = \Theta^l x_j^l \tag{16}$$

$$scores_{ij}^l = Sim\left(h_i^l, h_j^l\right) \tag{17}$$

$$\alpha_{ij}^l = Norm\left(scores^l\right) \tag{18}$$

$$h_i^{l+1} = \sigma\left\{\sum_{j\in\Gamma_i} \alpha_{ij}^l h_j^l\right\} \tag{19}$$

where σ denotes the non-linear activation function. Γ_i refers to the local neighborhoods of the given node v_i, which is exactly the first-order node in GAT. $Sim(\cdot)$ represents the alignment functions (listed in Table 1) and $Norm(\cdot)$ is the distribution functions, such as softmax, and sigmoid. The softmax function is defined in Eq. 9 and the sigmoid function is defined as:

$$sigmoid(x) = \frac{1}{1 + exp(-x)} \tag{20}$$

The alignment function in GAT with a single-layer feed-forward neural network parameterized by $\omega^T \in \mathbb{R}^{2\ dim}$ denotes as:

$$scores_{ij}^l = LeakyReLU\left(\omega^T \left[h_i^l \| h_j^l\right]\right) \tag{21}$$

where $[\cdot\|\cdot]$ refers to the operation of vector concatenation and $LeakyReLU(\cdot)$ represents an activation function. Compared with $ReLU(\cdot)$, $LeakyReLU(\cdot)$ introduces a constant parameter $\lambda \in (0, 1)$, defined as:

$$LeakyReLU(x) = \begin{cases} x, & if\ x > 0 \\ \lambda x, & if\ x \le 0 \end{cases} \tag{22}$$

The other widely used alignment function is a dot-product operation in self-attention, especially in Graph Transformers. The self-attention mechanism linearly projects the center node feature h_i^l to get the query vector and project the neighboring node features h_j^l to get the key and value vectors. The dot-product operation from the node view can be expressed as:

$$scores_{ij}^l = h_i^{l^T} h_j^l \tag{23}$$

To stabilize the learning process of self-attention, multi-head self-attention is also introduced into attention-based GNNs. While the multi-head attention aggregator can explore multiple representation subspaces between the center node and its neighborhoods (Brody et al. 2021). Every self-attention block contains its queries, keys, values, and learnable weight matrices, as shown in Fig. 4c. We summarize the two forms of multi-head self-attention blocks: concatenated multi-head and averaging multi-head. Multi-head self-attention (MHSA) with a concatenate operation can be formulated as:

$$h_i^{l+1} = MHSA\left(head_1, head_2, \cdots, head_K\right) = \|_{k=1}^{K} \sigma\left(\sum_{j\in\Gamma_i} \alpha_{ij}^{lk}\, \Theta^{lk} h_j^{lk}\right) \tag{24}$$

where $\|$ represents concatenation, α_{ij}^{lk} refers to the normalized attention coefficients computed by the k^{th} attention head in the l^{th} layer. σ is the non-linear activation function. And the averaging multi-head can be formulated as:

$$h_i^{l+1} = MHSA\left(head_1, head_2, \cdots, head_K\right) = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\Gamma_i} \alpha_{ij}^{lk}\, \Theta^{lk}\, h_j^{lk}\right) \tag{25}$$

Transformers are deep feed-forward artificial neural networks with a self-attention mechanism (Phuong and Hutter 2022). Similar to the traditional Transformer in Fig. 4d, Graph Transformers take multi-head self-attention as the core component. The Graph Transformer is introduced as a novel encoder-decoder architecture built with multiple blocks of self-attention, without convolution or recurrent modules (Yun et al. 2019). Each Transformer layer has two parts: multi-head self-attention modules and a position-wise feed-forward network (FFN) (Yang et al. 2022b). The generalized graph transformer with multi-head self-attention (MHSA) and feed-forward block (FFN) can be defined as:

$$z^{l+1} = LayerNorm\left(MHSA\left(h^l\right)\right) + h^l \tag{26}$$

$$h^{l+1} = LayerNorm\left(FFN\left(z^{l+1}\right)\right) + z^{l+1} \tag{27}$$

where z^{l+1} is the output of the first stage in the graph transformer layer. $LayerNorm(\cdot)$ is the layer normalization layer. h^{l+1} is the final output of this transformer block, the representations of the node in $(l+1)^{th}$ layer. The FFN layer with a non-linear activation function can be defined as:

$$FFN\left(h^l\right) = \sigma\left(h^l\, \Theta_1 + b_1\right)\Theta_2 + b_2 \tag{28}$$

3 Taxonomy of Attention-based GNNs

In this section, we outline a novel two-level taxonomy for attention-based GNNs from the perspective of development history and architectural perspectives. The upper level reveals the three developmental stages of attention-based graph neural networks, including Graph Recurrent Attention Networks (GRANs), Graph Attention Networks (GATs: intra-layer and inter-layer), and Graph Transformers. We summarize their characteristics in Table 2. The lower level focuses on various typical architectures of each stage, as shown in Fig. 6.

Figure 6 shows the two-level taxonomy of attention-based GNNs. The color in the figure represents the type of attention mechanism employed in a category.

GRANs: This kind of works focus on recurrent neural networks(RNNs) to learn representations of graph-structured data. GRANs are inspired by attention-based recurrent neural networks(RNNs) in deep learning (Liao et al. 2019). Therefore, we naturally divide these methods into two subclasses: GRU-Attention (Zhang et al. 2018) and LSTM-Attention (Xu et al. 2018b).

Intra-layer GATs: This kind of works introduce the attention mechanism into the local neighborhoods in the single-layer neural network with local attention. The intra-layer GATs (Veličković et al. 2018) usually place the local attention on the local neighborhoods in the graph within a single-layer neural network. Intra-layer GATs can be further divided into six subclasses, namely neighbor attention (Veličković et al. 2018), high-order attention (Yang et al. 2019), relation-aware attention (Li et al. 2020c; Busbridge et al. 2019), hierarchical attention (Wang et al. 2019d; Lin et al. 2022), attention sampling/pooling (Abu-El-Haija et al. 2018; Knyazev et al. 2019), and hyper-attention (Zhang et al. 2020b). Specifically, neighbor attention only considers the direct neighborhoods, while high-order attention takes k^{th}-hop neighborhoods and subgraphs into consideration. Relation-aware attention takes into account different types of relations. In addition to focusing on node-level attention, hierarchical attention also considers higher-level attention, such as path, group, and relationship. Attention sampling/pooling refers to attention-based GNNs used for sampling or pooling. Hyper-attention represents special attention mechanisms for hypergraphs.

Table 2 The characteristics of different developmental stages

Stages		Characteristics
GRANs		Local attention
		Limited by the long-term dependence bottleneck and order problem of RNN
GATs	Intra-layer	Local attention
		Stacking layers leads to high complexity and over-smoothing
	Inter-layer	Feature fusion attention
		Feature selection operation from different feature spaces, such as different levels, different channels, different views, or different time slices
Graph Transformers		Global attention
		Directly obtain high-order neighborhood information
		Catastrophic Spatio-temporal complexity and limited on large-scale graphs

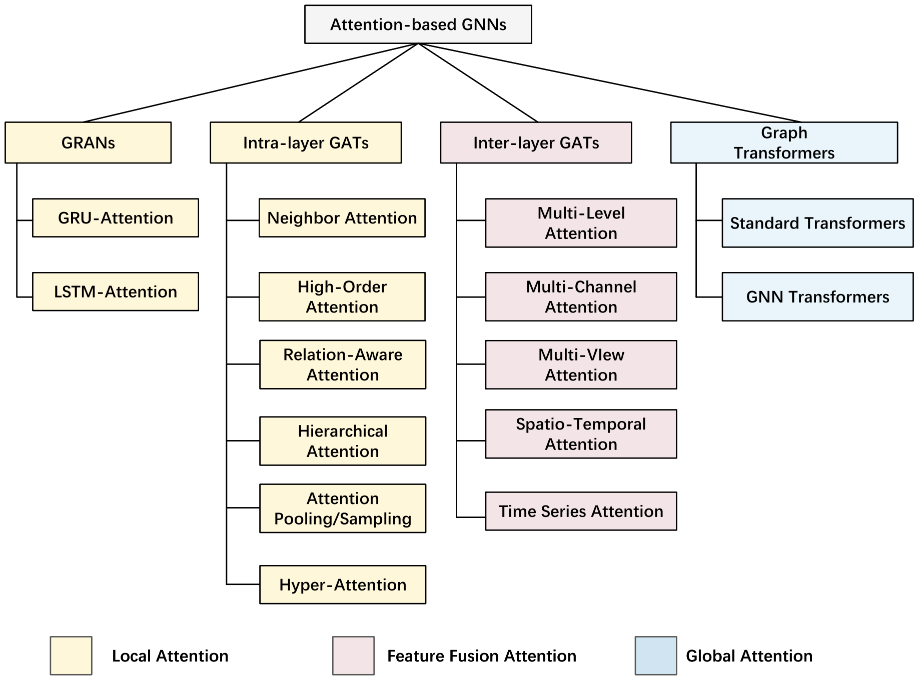


Fig. 6 Classification breakdown of methods for attention-based GNNs

Inter-layer GATs: This kind of works usually select features beyond neural network layers with multiple feature spaces, not just local neighborhoods. Across the neural network layer, attention in inter-layer GATs can be regarded as an operation of cross-layer fusion of different feature spaces with feature fusion attention. In this term, attention-based GNNs dynamically select features from different levels, different channels, different views, or different time slices. Therefore, we further divide these methods into five sub-categories (i.e., multi-level attention (Liu et al. 2020; Zhang et al. 2022c), multi-channel (Bo et al. 2021; Luan et al. 2021), multi-view (Wang et al. 2020b; Yuan et al. 2021b), Spatio-temporal attention (Sankar et al. 2018; Lu et al. 2019), and time series attention (Zhang et al. 2021c; Zhao et al. 2020)). By considering temporal attributes, Spatio-temporal attention usually uses time, spatial attention, or both in dynamic graphs, while time-series attention needs to construct dynamic graphs from time-series data first.

Graph Transformers: In the past two years, Transformers (Lin et al. 2021) have achieved superior performance in many tasks of NLP, CV, and GRL. Graph Transformers generalize the Transformer architecture to graph representation learning, capturing long-range dependency (Ying et al. 2021). Different from previous methods with local attention, Graph Transformers learn higher-order graph properties directly via global attention. Graph Transformers have developed rapidly in the field of graph deep learning, especially in the task of graph classification on small and medium-sized graphs. We further divide Graph Transformers into two sub-categories, namely standard Transformers (Ying et al. 2021) and GNN Transformers (Nguyen et al. 2019). Standard Transformers usually utilize the self-attention mechanism to all nodes of the input graph, ignoring adjacencies between nodes, while GNN Transformers use the GNN layer to obtain adjacency information.

We tabularize representative works for each sub-category in the different stages as shown in Table 3. In the following sections, we detail each representative work in terms of motivation, characteristics, or functions

4 GRANs

GRANs are inspired by attention-based recurrent neural networks (RNNs). Table 3 shows the representative works of two kinds of GRANs, namely GRU-Attention and LSTM-Attention.

4.1 GRU-attention

Gated Recurrent Units (GRU) can be regarded as operators acting on the current input and previous state to control how much of the input should be taken into account, and how much past information should be remembered (or forgotten) in the computation of the new state (Ruiz et al. 2020).

Based on the previous work on GNNs (Scarselli et al. 2008), GGNN introduces GRU with a soft attention mechanism. The graph representation in GGNN uses context to focus attention on which nodes are important to the current graph-level task (Li et al. 2016). Further, GRNN proposes a general learning framework leveraging the notion of a recurrent hidden state together with graph signal processing (GSP) (Ruiz et al. 2020). Under this framework, GRNN generates meaningful representations of graph signals by incorporating the importance of a node's features to its neighbors with three different gating mechanisms: time, node, and edge gates.

Unlike the traditional multi-head attention mechanism, which equally computes all attention heads, GaAN uses a convolutional sub-network to control each attention head's importance (Zhang et al. 2018). GaAN assigns different importance to each head through computing an additional soft gate mechanism between 0 and 1, 0 for low importance and 1 for high importance (Zhang et al. 2018). Similar to the graph generation task in GaAN, GRAN captures the auto-regressive conditioning between the already-generated and to-be-generated parts of the graph using GNNs with attention, which not only reduces the dependency on node ordering but also bypasses the long-term bottleneck caused by the sequential nature of RNNs (Liao et al. 2019).

4.2 LSTM-attention

Another classic and simple RNN structure is Long Short-Term Memory (LSTM). GraphSAGE (Hamilton et al. 2017) first introduces the LSTM in GNNs as an aggregator, different from mean and pooling aggregators. Cooperating with sampling, GraphSAGE aggregates the messages from the local neighborhood of the node through the aggregator. In the same way, JK-Net (Xu et al. 2018b) adapts LSTM-Attention as a layer-aggregation to aggregate the jumping representations from the previous layers. LSTM-Attention in JK-Net is node adaptive because the attention scores are different for each node (Xu et al. 2018b).

Due to the ability to capture long-range dependencies, GAM adopts an LSTM and proposes a solution for graph classification based on attention-guided walks (Lee et al. 2018). The attention mechanism in GAM focuses on small but informative parts of the graph and avoids noise in the rest of the graph. GeniePath proposes an adaptive path layer with two

Table 3 Categories of each developmental stage and representative works

Categories		Representative works
GRANs	GRU- Attention	GGNN (Li et al. 2016), GRNN (Ruiz et al. 2020), GaAN (Zhang et al. 2018), GRAN (Liao et al. 2019)
	LSTM- Attention	JK-Net (Xu et al. 2018b), GAM (Lee et al. 2018), GeniePath (Liu et al. 2019b)
Intra-layer GATs	Neighbor Attention	GAT (Veličković et al. 2018), DMP (Yang et al. 2021b), SuperGAT (Kim and Oh 2021), CPA (Zhang and Xie 2020), GATv2 (Brody et al. 2021), PPRGAT (Choi 2022), Simple-HGN (Lv et al. 2021), DPGAT (Brody et al. 2021), C-GAT (Wang et al. 2019a), GANet (Gao and Ji 2019), CAT (He et al. 2021), MSNA (Wang et al. 2020c), AGNN (Thekumparampil et al. 2018), HTNE (Zuo et al. 2018), HAT (Zhang et al. 2021d), HGCN (Chami et al. 2019), Hype-HAN (Zhang and Gao 2021), DKGAT (Yang et al. 2020b), DAGL (Mou et al. 2021), SCGA (Kim et al. 2021c), Co-GAT (Qin et al. 2021a), ED-GAT (Ma et al. 2020), TD-GAT (Huang and Carley 2019), GATON (Yang et al. 2020a), GRAM (Choi et al. 2017)
	High-Order Attention	SPAGAN (Yang et al. 2019), PaGNN (Yang et al. 2021d), CGAT (Cao et al. 2020), ADSF (Zhang et al. 2019), MAGNA (Wang et al. 2021), T-GAP (Jung et al. 2021a)
	Relation-Aware Attention	SiGAT (Huang et al. 2019b), SNEA (Li et al. 2020c), RGAT (Busbridge et al. 2019), WRGNN (Suresh et al. 2021), HetSANN (Hong et al. 2020), EAGCN (Shang et al. 2018), TALP (Li et al. 2020b), KGAT (Wang et al. 2019c), GATNE (Cen et al. 2019), CGAT (Lin and Wang 2020), RelGNN (Qin et al. 2021b), AFE (Nathani et al. 2019), DisenKGAT (Wu et al. 2021a), GTAN (Zhang et al. 2021b), R-GAT (Wang et al. 2020a), ReGAT (Li et al. 2019), AD-GAT (Cheng and Li 2021)
	Hierarchical Attention	HAN (Wang et al. 2019d), PSHGAN (Mei et al. 2022), PRML (Zhao et al. 2017), GraphHAM (Lin et al. 2022), LAN (Wang et al. 2019b), RGHAT (Zhang et al. 2020c), DANSER (Wu et al. 2019b), UVCAN (Liu et al. 2019a), GCATSL (Long et al. 2021), DAGC (Sun et al. 2020), AGCN (Li et al. 2020a), HGAT (Yang et al. 2021e)
	Attention Sampling/Pooling	GAW (Abu-El-Haija et al. 2018), NLGCN (Liu et al. 2021), SAGPool (Lee et al. 2019a), Attpool (Huang et al. 2019a), ChebyGIN (Knyazev et al. 2019)
	Hyper-Attention	Hyper-SAGNN (Zhang et al. 2020b), HHGR (Zhang et al. 2021a), HyperTeNet (Vijaikumar et al. 2021), Hyper-GAT (Bai et al. 2021)

Table 3 (continued)

Categories		Representative works
Inter-Layer GATs	Multi-Level Attention	DAGNN (Liu et al. 2020), TDGNN (Wang and Derr 2021), GAMLP (Zhang et al. 2022c)
	Multi-Channel Attention	FAGCN (Bo et al. 2021), ACM (Luan et al. 2021)
	Multi-View Attention	AM-GCN (Wang et al. 2020b), MV-GNN (Yuan et al. 2021b), GENet (Tang et al. 2021), UAG (Feng et al. 2021), MVE (Qu et al. 2017), MGAT (Tao et al. 2020)
	Spatio-Temporal Attention	DySAT (Sankar et al. 2018), TemporalGAT (Fathy and Li 2020), GAEN (Shi et al. 2021a), MMDNE (Lu et al. 2019), TGAT (Xu et al. 2020), T-GNN (Xu et al. 2022), ST-GCN (Yan et al. 2018), GMAN (Zheng et al. 2020), ASTGCN (Guo et al. 2019), ConSTGAT (Fang et al. 2020)
	Time Series Attention	RainDrop (Zhang et al. 2021c), MTAD-GAT (Zhao et al. 2020), GACNN (He and Shin 2020)
Graph Transformers	Standard Transformers	Graphormer (Ying et al. 2021), HOT (Kim et al. 2021b), PAGAT (Chen et al. 2019), GTA (Seo et al. 2021), GT (Dwivedi and Bresson 2020), SAN (Kreuzer et al. 2021), GraphBert (Zhang et al. 2020a), UniMP (Shi et al. 2021b)
	GNN Transformers	GROVER (Rong et al. 2020), UGformer (Nguyen et al. 2019), GMT (Baek et al. 2021), GTN (Yun et al. 2019), GraphFormers (Yang et al. 2021a), HGT (Hu et al. 2020b), GTOS (Cai and Lam 2020), Graph-Writer (Koncel-Kedziorski et al. 2019), KHGT (Xia et al. 2021b), GATE (Ahmad et al. 2021), STAGIN (Kim et al. 2021a)

complementary functions: breadth function and depth function (Liu et al. 2019b). The adaptive breadth function learns the importance of different sized neighborhoods and adaptively selects a set of important 1-hop neighbors with a parameterized generalized linear attention operator. While the adaptive depth function for depth exploration can extract useful signals and filter noisy signals up to long-distance neighbors with a gated unit. Even though LSTMs have a more sophisticated memory model when compared to simple RNNs, it has been shown that they still have trouble remembering information that was inputted too far in the past (Lee et al. 2018).

5 Intra-Layer GATs

Considering different local neighborhoods and different functions, intra-layer GATs can be further divided into six subclasses including neighbor attention, high-order attention, relation-aware attention, hierarchical attention, attention sampling/pooling, and hyper-attention. Table 3 shows the representative works of each subclass.

5.1 Neighbor Attention

As the most famous attention network, GATs are widely used in a variety of graph-structured data scenes. GATs with neighbor attention compute the hidden representations of each node in the graph by attending to its neighbors in the local neighborhoods. GAT takes the lead in introducing the attention mechanism into graph neural networks to aggregate representations of local neighbor nodes in the graph (Veličković et al. 2018). Every node in GAT attends to the neighbors in its neighborhood and treats its representation as a query (Brody et al. 2021). GAT takes a single-layer feed-forward neural network (FFN) as an alignment function to calculate the attention scores:

$$scores_{ij}^{l} = LeakyReLU\left(\omega^{T}\left[\Theta\ h_{i}^{l} \| \Theta h_{j}^{l}\right]\right) \tag{29}$$

where $scores_{ij}^{l}$ indicates the importance of the node v_j to node v_i in the l^{th} layer. As an initial step, the FFN adopts a shared linear transformation, parametrized by a weight matrix Θ, to perform feature transformation for every node in the graph. With a learnable weight vector $\omega \in \mathbb{R}^{2\ dim}$, the FNN applies the LeakyReLU as a non-linear activation function. Ignoring unconnected node pairs, GAT only computes the masked attention for each node with its first-order neighbors via local attention, as shown in Fig. 5a. Subsequently, GAT normalizes the attention scores using the softmax function:

$$\alpha_{ij} = softmax_{j}\left(scores_{ij}\right) = \frac{exp\left(scores_{ij}\right)}{\sum_{k\in\Gamma_{i}} exp\left(scores_{ik}\right)} \tag{30}$$

where v_j and v_k are the neighbor nodes in the neighborhood Γ_i of node v_i. Different from the non-negative distribution function via a softmax non-linear function in GAT, DMP (Yang et al. 2021b) takes tanh as a non-linear function, whose output is zero-centered and ranged in $(-1,\ 1)$, defined as:

$$\alpha_{ij} = tanh\left(scores_{ij}\right) = \frac{exp\left(scores_{ij}\right) - exp\left(-scores_{ij}\right)}{exp\left(scores_{ij}\right) + exp\left(-scores_{ij}\right)} \tag{31}$$

In DMP, the positive weights correspond to low-passing filtering capturing the similarity between nodes, while the negative weights facilitate the filtering of high-frequency to reduce the difference (Bo et al. 2021; Yang et al. 2021b).

In addition, GAT adapts the multi-head attention to stabilize the learning process of self-attention. A concatenated multi-head concatenates the output representations of different heads. An averaging multi-head is employed to average the representations of different heads in the final layer as output for the prediction task.

The attention function in GAT always weighs one key at least as much as any other key, unconditioned on the query (Brody et al. 2021). To address this limitation, GATv2 (Brody et al. 2021) makes a simple fix by modifying the order of operations in the alignment function, defined as:

$$scores^{l}_{ij} = \omega^{T}\, LeakyReLU\left(\Theta\left[\, h^{l}_{i} \| h^{l}_{j}\right]\right) \tag{32}$$

GATv2 performs an additional empirical comparison to DPGAT (Brody et al. 2021), applying the scaled dot-product attention of the Transformer (Vaswani et al. 2017):

$$scores^{l}_{ij} = \frac{(h^{l^T}_{i}\Theta_{Q})(h^{l^T}_{j}\Theta_{K})^{T}}{\sqrt{dim}} \tag{33}$$

PPRGAT (Choi 2022) incorporates the Personalized PageRank (PPR (Klicpera et al. 2018)) information into the GAT and GATv2, which utilizes the full potential of GATs. For this, the alignment functions in GAT and GATv2 are modified as:

$$scores^{l}_{ij} = LeakyReLU\left(\omega^{T}\left[\Theta\, h^{l}_{i} \| \Theta h^{l}_{j} \| \Pi_{ij}\right]\right) \tag{34}$$

$$scores^{l}_{ij} = \omega^{T}\, LeakyReLU\left(\Theta\left[\, h^{l}_{i} \| h^{l}_{j} \| \Pi_{ij}\right]\right) \tag{35}$$

where Π_{ij} is the PPR matrix in the preprocess step before starting the training. Simple-HGN (Lv et al. 2021) also adopts the same method for feature concatenation with the edge relationship type between the node v_i and v_j.

Instead of the soft graph attention operator (GAO) in GAT, GANet (Gao and Ji 2019) introduces the hard attention operator (HGAO) and channel-wise graph attention operator (CGAO) with dot product attention. HGAO uses the hard attention mechanism by attending to only important nodes, while CGAO avoids the dependency on the adjacency matrix, leading to dramatic reductions in computational resource requirements (Gao and Ji 2019). Besides considering the layer-wise node features propagated within the GNN, CAT learns representations of nodes in the graph with conjoint attention, considering additional structural information, such as node cluster embedding and higher-order structural correlations when computing attention scores (He et al. 2021). MSNA (Wang et al. 2020c) extracts comprehensive and expressive neighborhood features with two neighborhood attention including self neighborhood attention network (SNAN) and cross neighborhood attention network (CNAN). The SNAN predicts the link of two nodes by encoding and matching their respective neighborhood information, while the CNAN with a cross neighborhood attention directly captures structural interactions between two nodes (Wang et al. 2020c).

To alleviate the over-fitting problem, C-GAT (Wang et al. 2019a) proposes two additional margin-based constraints as loss functions on GAT. CPA preserves cardinality information in attention-based aggregation, which can be applied to any kind of attention mechanisms (Zhang and Xie 2020). To improve the graph attention model for noisy graphs, SuperGAT (Kim and Oh 2021) exploits two commonly used attention mechanisms: a single-layer feed-forward neural network (FFN) in GAT and a dot product (DP), for a self-supervised task to predict edges (Kim and Oh 2021).

Apart from FNN and dot product, there are a series of classical attention-based GNNs adopting other alignment functions to calculate attention scores, such as cosine similarity and Euclidean distance. AGNN (Thekumparampil et al. 2018) removes all the intermediate fully-connected layers and replaces the propagation layers with an attention mechanism that computes attention with cosine similarity. HTNE chooses a negative Euclidean distance function to score the affinity between the source and history node to better determine the influence of historical neighbors on the current neighbors of a node (Zuo et al. 2018).

Beyond Euclidean space, HAT (Zhang et al. 2021d) attempts the GAT with an attention mechanism in hyperbolic space (Yang et al. 2022a). To calculate the attention, HAT transforms the features in a graph into gyrovector space and then proposes the attention-based hyperbolic proximity to aggregate the features (Zhang et al. 2021d). HGCN also introduces a hyperbolic attention-based aggregation scheme based on the Riemannian manifold, to learn inductive node representations for hierarchical and scale-free graphs (Chami et al. 2019). Hyperbolic attention operation makes use of hyperbolic geometry in both the computation of the attention weights and the aggregation operation (Gulcehre et al. 2018). Further, Hype-HAN is a hierarchical embedding method based on three types of hyperbolic manifolds on Riemannian geometries (Yang et al. 2022a), including the Lorentz model, Klein model, and Poincaré model, for text classification tasks (Zhang and Gao 2021).

Up to the present, a great variety of graph attention networks and their variants have been proposed in many artificial intelligence fields, such as natural language processing (Qin et al. 2021a; Ma et al. 2020; Huang and Carley 2019; Yang et al. 2020a), computer vision (Yang et al. 2020b; Mou et al. 2021; Kim et al. 2021c), and medical health (Choi et al. 2017). In the field of natural language processing, we can think of words as nodes and then construct graph models. GNNs with neighbor attention utilize the dependency relationship among words for sentiment analysis (Qin et al. 2021a; Ma et al. 2020; Huang and Carley 2019), and topic modeling (Yang et al. 2020a). GRAM proposes a graph-based attention model for healthcare representation learning (Choi et al. 2017). For computer vision, many works have extended neighbor attention to extracting local features in images or videos, e.g., 3D object recognition (Yang et al. 2020b), image restoration (Mou et al. 2021), and video-grounded dialogue (Kim et al. 2021c).

5.2 High-order attention

Neighbor attention often only focuses on aggregating information from first-order neighbors within each layer. To obtain the representations of higher-order neighbors, the conventional approach is stacking multiple layers, which also brings high complexity and over-smoothing (Yang et al. 2019). Such an attention mechanism is limited because it does not consider nodes that are not connected by an edge, which could provide important contextual information. In Fig. 7, high-order attention (path-based and *k*-hop neighbors)

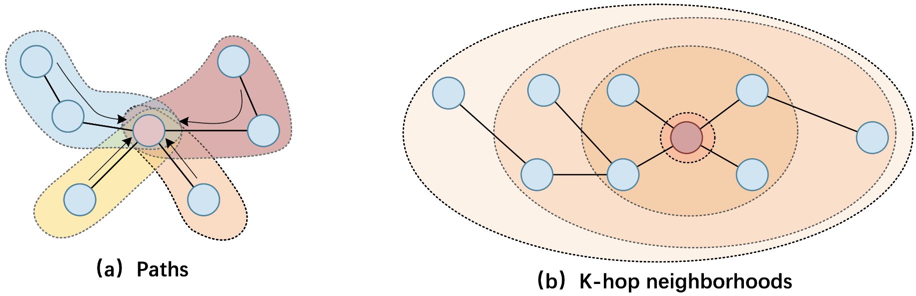


Fig. 7 Higher-order message-passing along paths and *k*-hop neighborhoods. Path-based message-passing provides a mechanism for aggregating messages along meta-paths, while *k*-hop neighborhoods aggregate messages from different hop neighborhoods

aggregates information from distant neighbors and explores the graph topology, all in a single neural network layer(Yang et al. 2019).

To account for higher-order neighbors, one can stack multiple layers to enlarge the size of the receptive field, horribly bringing high complexity. The other is high-order neighbors in a single neural network layer, achieved by shortest paths, to capture the more global graph topology. Unlike conventional neighbor attention that carries out node-based attention within each layer, SPAGAN proposes path-based attention accounting for the influence of a sequence of nodes, or shortest path, between the current node and its higher-order neighbors (Yang et al. 2019). PaGNN (Yang et al. 2021d) develops a novel path-aware GNN for link prediction, integrating inter-action and neighborhood information via broadcasting and aggregating operations. To effectively preserve the structural topology and semantic properties in heterogeneous information networks, CGAT (Cao et al. 2020) adopt multiple meta-paths-based sampling and pre-training process with pair-wise attention.

To fully exploit rich, high-order structural details in GATs, ADSF (Zhang et al. 2019) proposes an adaptive structural fingerprints model. The key idea of ADSP is to update the representation of each node within a local receptive field consisting of its high-order neighbors. Modeling attention flow on graphs is another way to obtain higher-order information, which effectively contributes to the information flow implemented through message passing (Xu et al. 2018c). To eliminate noisy high-frequency information from the graph, MAGNA (Wang et al. 2021) captures large-scale structural information in each layer with a low-pass filter. MAGNA propagates the attention scores across the graph with Personalized PageRank, increasing the receptive field for each layer of the GNN (Wang et al. 2021). Similarly, T-GAP models a path traversal with the soft approximation of attention flow, iteratively propagating the attention value of each node to its outgoing neighbor nodes (Jung et al. 2021a).

5.3 Relation-aware attention

Most of the interactions in social networks are positive relationships, such as friendship, following, and support. Meanwhile, some negative links exist in the real world indicating disapproval, disagreement, or distrust (Huang et al. 2019b). However, some of the graphs are signed graphs with both positive and negative links in Fig. 8a. GAT is designed to graph only considering positive links and ignoring the negative ones in the signed graph. Signed Graph Attention Networks (SiGAT) generalizes GAT to signed graphs (Huang

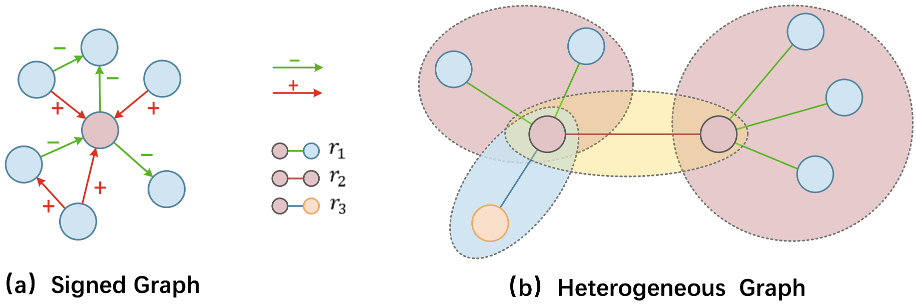


Fig. 8 Relation-aware neighborhoods in signed graph and heterogeneous graph. **a** There are two types of edges in the signed graph: positive and negative. **b** Different color blocks represent different types of relationships

et al. 2019b), which incorporates graph motifs into GAT to capture two famous theories in the signed graph, i.e., balance and status theory. In SiGAT, different motifs represent different relation-aware influences, when aggregating and propagating messages on the signed graph to generate node embeddings via an attention mechanism (Huang et al. 2019b). Also in the signed graph, SNEA computes the importance coefficient for pair nodes connected by different types of links when aggregating the embedding with a self-attention mechanism (Li et al. 2020c).

More complex than signed networks, heterogeneous information networks (HINs) are also ubiquitous in our daily life. As shown in Fig. 8b, HINs usually consist of various types of vertices connected by various types of relations (Zheng et al. 2022; Wang et al. 2022). Taking into consideration edge type $r \in R$, the attention function on HINs to calculate the attention scores can be defined as:

$$scores_{ij}^{r} = LeakyReLU\left(\omega^{T}\left[\Theta_{r} h_{i}^{r} \| \Theta_{r} h_{j}^{r}\right]\right) \tag{36}$$

$$\alpha_{ij}^{r} = \frac{exp\left(scores_{ij}^{r}\right)}{\sum_{k \in \Gamma_{i}^{r}} exp\left(scores_{ik}^{r}\right)} \tag{37}$$

$$h_{i}^{l+1} = \sigma\left\{\sum_{r \in R} \sum_{j \in \Gamma_{i}^{r}} \alpha_{ij}^{r} \Theta_{r} h_{j}^{l}\right\} \tag{38}$$

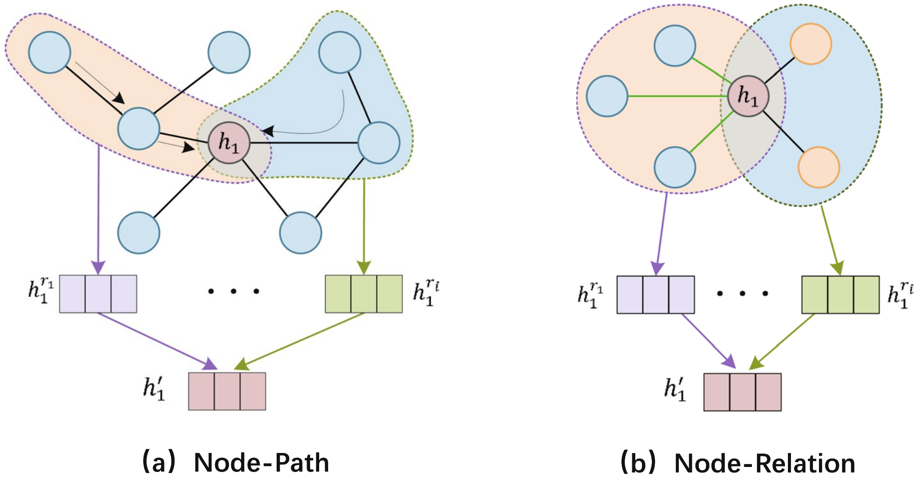
where $r \in R$ is the type of edge, Γ_{i}^{r} refers to the neighborhood of the node v_i with the edge type r.

Different relations convey distinct pieces of information. Relation-aware graph attention networks aggregate information with a masked self-attention that takes account of local relational structure as well as node features. Considering the relation between nodes in HINs, RGAT proposes two variants, Within-Relation Graph Attention (WIRGAT) and Across-Relation Graph Attention (ARGAT), under an additive and multiplicative logit construction (Busbridge et al. 2019). HetSANN (Hong et al. 2020) aggregates multi-relational information in neighborhoods with two kinds of attention scoring functions. In HetSANN, the type-aware attention layer adopts the voices-sharing product. WRGAT (Suresh et al. 2021) transforms the input graph into a computation graph containing both proximity and structural information with the different types of edges. The generated multi-relational graph has a high-level assortativity and preserves rich structural information from the original graph (Suresh et al. 2021). EAGCN proposes an edge attention-based multi-relational GCN on chemical graphs with multiple relationships (Shang et al. 2018). To avoid the fusion vector ignoring the effects of the local type information, TALP models the effect of type information and fusion information from local and global perspectives simultaneously (Li et al. 2020b), based on a two-layer graph attention architecture. KGAT adaptively aggregates the embeddings from neighbors of the current node and updates the node's representation with an attention mechanism to distinguish the importance of the neighbors (Wang et al. 2019c). To capture the influential information in different edge types, GATE formalizes the attributed multiplex heterogeneous network embedding problem as well as uses the attention mechanism (Cen et al. 2019).

In particular, RelGNN (Qin et al. 2021b) generates the states of different relations and leverages them along with the node states to weigh the messages. Further, RelGNN balances the importance of attribute features and topological features via a self-attention mechanism and then generates the final representations of nodes (Qin et al. 2021b). To learn both node and topic embeddings while preserving the graph structural information, CGAT (Lin and Wang 2020) guides information aggregation via a channel-aware attention mechanism focusing on the edges. Relational graph attention networks play a significant role in some specific scenarios with rich relational structure information, such as knowledge graph (Nathani et al. 2019; Wu et al. 2021a; Zhang et al. 2021b), NLP (Wang et al. 2020a), CV (Li et al. 2019), and stock prediction (Cheng and Li 2021). Since knowledge graphs are inherently graph-structured data with multiple relationship types, relation-aware attention is well-suited for tasks in knowledge graphs, such as relation prediction (Nathani et al. 2019), knowledge graph completion (Wu et al. 2021a), and question answering (Zhang et al. 2021b).

5.4 Hierarchical attention

Heterogeneous information networks (HINs) contain rich semantic information and hierarchical information, such as different types of nodes and links (Wang et al. 2022). In Fig. 9a, different meta-paths in HINs may extract diverse semantic information, while the relation-based hierarchical attention is illustrated in Fig. 9b. Based on meta-paths, HAN (Wang et al. 2019d) designs a heterogeneous graph neural network with hierarchical attention including node-level and semantic-level attention. The node-level attention learns the importance between a node and its meta-path-based neighbors, while the semantic-level attention learns the importance of different meta-paths (Wang et al. 2019d). Then HAN generates node representations by aggregating the received information from meta-path-based neighbors in a hierarchical manner. Similarly, PSHGAN (Mei et al. 2022) first learns the weights of two nodes in the meta-path or meta-structure via a local attention mechanism. Then, PSHGAN learns a global attention weight based on meta-paths and meta-structures. In the end, PSHGAN uses computed dual-level attention to aggregate and update the representations of nodes. PRML (Zhao et al. 2017) focuses on both node-level and path-level attention proximity of the endpoints based on their betweenness paths to discriminate the representations.



(a) Node-Path

(b) Node-Relation

Fig. 9 Hierarchical attention with node-path and node-relation. In addition to node-level attention, hierarchical attention often has higher-level attention, such as path, and relation

RGHAT, equipped with a hierarchical attention mechanism including entity- and relation-level attention, can effectively aggregate the local neighborhood information of each entity (Zhang et al. 2020c). The entity-level attention highlights the importance of different neighboring entities under the same relation, while the relation-level attention is inspired by the intuition that different relations have different weights for indicating an entity (Zhang et al. 2020c). LAN aggregates neighbor information with both rules- and network-based attention weights (Wang et al. 2019b) and focuses on both neighborhood features and query relations. GraphHAM (Lin et al. 2022) aggregates neighboring states to generate node embeddings with group-level and individual-level attention for graph embedding. GraphHAM captures the information from neighborhoods of different scopes by stacking multiple layers, where the node states output by a lower layer are used as input to the layer above it (Lin et al. 2022).

Hierarchical attention-based GNNs are widely used in recommender systems (Wu et al. 2019b; Liu et al. 2019a), synthetic lethality (Long et al. 2021), point cloud (Sun et al. 2020), action detection (Li et al. 2020a), and short text classification (Yang et al. 2021e).

5.5 Attention sampling/pooling

Grap sampling is an effective method to select representative nodes from a large number of nodes (Hamilton et al. 2017), as shown in Fig. 10a. That is often adopted for large-scale graphs to improve efficiency (Zeng et al. 2019). GraphSAGE (Hamilton et al. 2017) first learns an aggregator function that generates embeddings by sampling and aggregating features from a node's local neighborhood. Taking account of the attention mechanism, GAW (Abu-El-Haija et al. 2018) guides the random walk to optimize an upstream objective via the proposed attention model on the power series of the transition matrix. Furthermore, the attention mechanism in GAW only guides the random walk with a softmax function under the learning procedure (Abu-El-Haija et al. 2018). To distinguish the importance of nodes, NLGAT (Liu et al. 2021) proposes a simple yet effective non-local aggregation framework with efficient attention-guided sorting for GNNs.

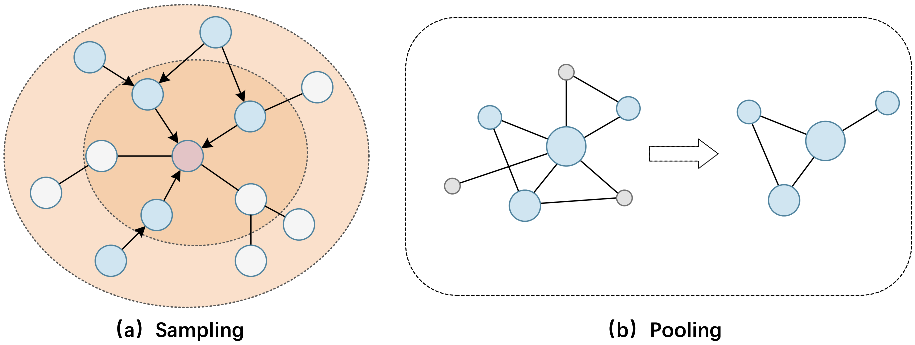


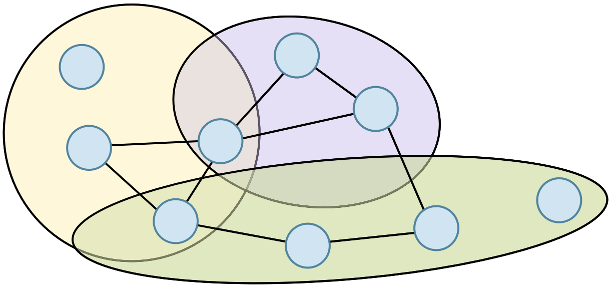
Fig. 10 Sampling and Pooling in GNNs. Sampling is often used when messages are aggregated, while pooling is often used for readout operations for graph classification tasks

To generalize GNNs to larger, more complex, or noisy graphs, graph pooling aims to remove any number of nodes, so that the receives smaller graph turns into an input graph in the following layer, such as DiffPool (Ying et al. 2018). For attention-based graph pooling in Fig. 10b, the attention mechanism in the local neighborhood could distinguish the nodes that should be dropped and the nodes that should be retained. SAGPool (Lee et al. 2019a) is a graph pooling method based on self-attention. Self-attention in SAGPool allows the proposed pooling method to consider both node features and graph topology (Lee et al. 2019a). To learn hierarchical representation for graph embedding, AttPool (Huang et al. 2019a) selects nodes that are significant for graph-level representation adaptively and then generates hierarchical features via the attention-based aggregator. ChebyGIN designs simple graph reasoning tasks to study the attention in GNNs under a controlled environment, and results show that attention can make GNNs more robust to larger and noisy graphs (Knyazev et al. 2019).

5.6 Hyper-attention

Modeling complex relationships between objects, hypergraphs exploit the high-order relationship and local clustering structure by hyperedges beyond a pairwise formulation (Zhang et al. 2022b), as shown in Fig. 11. Recently, Hypergraphs with high-order relationships and variable hyperedges, attract much attention from researchers focusing on graph representation learning (Zhang et al. 2022a).

Fig. 11 HyperGraph. Different color blocks represent different Hyper-edges. Hyper-edges are the interactions that occur between multiple nodes



Considering the complex relationships in hypergraphs, Hyper-SAGNN (Zhang et al. 2020b) develops a new GNN model for the graph representation learning of general hypergraphs with various hyperedges. Since group members have different importance, HHGR (Zhang et al. 2021a) adopts a weighted sum function to generate the attentive group representation under a double-scale self-supervised setting. The hierarchical hypergraph convolutional network in HHGR consists of an attention-based group aggregator, which captures the user interactions within and beyond groups by propagating information from the user level to the group level (Zhang et al. 2021a). To learn the multi-hop relationship among the nodes in hypergraphs, HyperTeNet (Vijaikumar et al. 2021) designs a self-attention-based hypergraph neural network to learn the ternary relationships among the interacting nodes in a 3-uniform hypergraph. Hyper-GAT enhances the capacity of representation learning in high-order encoded by hyperedges with an attention module (Bai et al. 2021).

6 Inter-layer GATs

Across the neural network layer, inter-layer GATs combine representations from different feature spaces via feature fusion methods. According to different fusion methods, we divide these attention-based GNNs into five sub-categories, including multi-level attention, multi-channel attention, multi-view attention, Spatio-temporal attention, and time series attention, as listed in Table 3.

6.1 Multi-level attention

Through stacking neural network layers, GNNs could learn node representations by aggregating the information from the multi-hop neighborhoods (Zhang et al. 2022c). One layer of neighborhood aggregation in GNNs only considers immediate neighbors, but the performance decreases when going deeper to enable larger receptive fields due to over-smoothing (Liu et al. 2020). Repeated propagation and aggregation make node representations indistinguishable from different classes (Li et al. 2018). To overcome the limitation of over-smoothing, some work has attempted to improve GNNs (Xu et al. 2018b; Liu et al. 2021; Li et al. 2018). Among them, some works adaptively select multi-level features cross-layer with an attention mechanism, as shown in Fig. 12.

DAGNN (Liu et al. 2020) transforms and propagates the representations of nodes with the ability to capture information from large and adaptive receptive fields. Specifically,

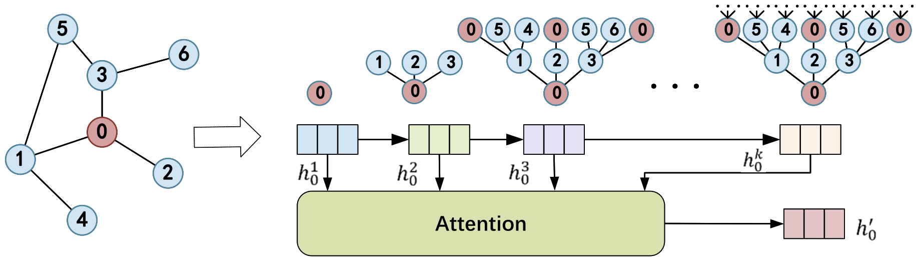


Fig. 12 Multi-level attention with different receptive fields. For central node v_0, its final representation is a combination of representations from different levels of neighborhoods through an attention mechanism

DAGNN decouples transformation and propagation to leverage large receptive fields with multi-hop neighborhoods. For each node, DAGNN balances the information from local and global neighborhoods with an attention mechanism, thus leading to more discriminative node representations (Liu et al. 2020). To alleviate feature smoothing among different layers, TDGNN (Wang and Derr 2021) further disentangles neighborhoods in different layers with a tree decomposition process. TDGNN flexibly aggregates information from large receptive fields utilizing a deeper multi-hop dependency with graph diffusion (Wang and Derr 2021). To better combine the propagated features, GAMLP (Zhang et al. 2022c) adopts three receptive field attention mechanisms including smoothing attention, recursive attention, and JK attention.

6.2 Multi-channel attention

Generally, most existing GNNs can be seen as low-pass filter, that updates node representations by aggregating information from neighbors in neighborhoods (Wu et al. 2019a). In the low-pass filter, GNNs make use of node features as low-frequency signals, which are based on the assortative assumption that the node tends to connect with similar nodes. Sometimes, the useful high-frequency signals that capture the difference between nodes are mixed or ignored (Bo et al. 2021). Different nodes in the graph may have different needs for the information in the different channels (frequency) (Luan et al. 2021), as shown in Fig. 13.

FAGCN (Bo et al. 2021) adaptively aggregates signals with different frequencies during message passing. FAGCN first designs an experimental study about low-frequency and high-frequency signals, where the results show that exploring low-frequency signals only is distant from learning an effective node representation in different scenarios (Bo et al. 2021). Based on this observation, FAGCN presents a novel frequency adaptation graph convolutional network via an attention mechanism to adaptively combine the low-frequency and high-frequency signals (Bo et al. 2021). Also considering different signals, ACM (Luan et al. 2021) proposes a framework to adaptively exploit aggregation,

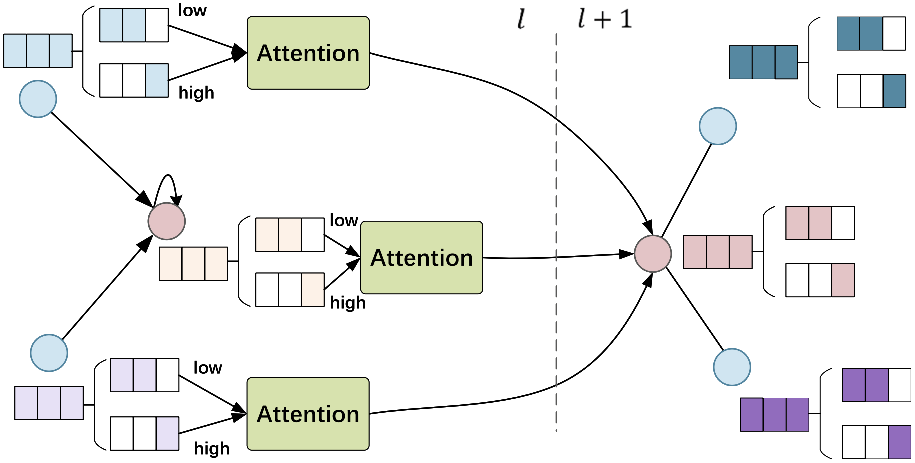


Fig. 13 Multi-channel attention with high- and low-frequency signals. Left: Low- and high-frequency information is aggregated by attention-weighted. Right: Update the representation of the central node based on messages from neighborhoods and itself in the upper layer

diversification, and identity channels to address harmful heterophily. To adaptively aggregate information from different channels, ACM learns node-wise attention to combine the different signals in three channels (Luan et al. 2021).

6.3 Multi-view attention

Multi-view graph representation learning learns representations of nodes in the graphs with multiple views, such as different topologies, which aims to generate more robust node representations from different views (Yuan et al. 2021b). The multi-view GNNs usually integrate the embeddings from multiple feature spaces of different views to update the final node representations, and not all views aggregator subspaces are equally important. As shown in Fig. 14, the attention mechanism in multi-view GNNs learns the adaptive importance weights of the embeddings from different views, to aggregate the view-specific node representations on each view (Yuan et al. 2021b).

To combine topological structures and node features substantially, AM-GCN (Wang et al. 2020b) selects the specific and common representations from different views including topological structures, node features, and their combinations simultaneously. AM-GCN first constructs a new graph topology based on features, named feature graph. With the feature graph and the topology graph, AM-GCN adaptively learns the deep correlation information in the feature space, topology space, and both of them, via an attention mechanism (Wang et al. 2020b). UAG develops a Bayesian Uncertainty Technique (BUT) to explicitly

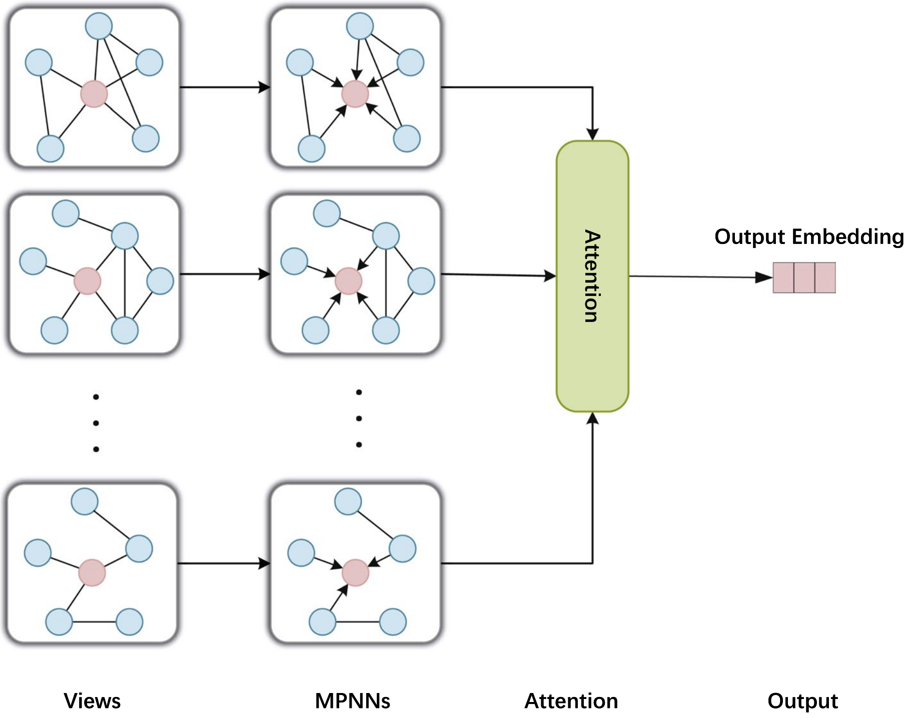


Fig. 14 Multi-View attention. Construct different graph structures for message passing and then use attention for message fusion

capture uncertainties in GNNs and further employs an Uncertainty-aware Attention Technique (UAT) (Feng et al. 2021). The UAT in UAG defends the adversarial attack on GNNs by assigning less impact on nodes with high uncertainty, thus, mitigating their impact on the final prediction (Feng et al. 2021).

Based on the three views reflecting local, global structure, and feature similarity of nodes, MV-GCN (Yuan et al. 2021b) designed an attention-based strategy to fuse the node representations from different views. GENet (Tang et al. 2021) introduces ensemble learning into GNNs based on different views with a drop-edge mechanism to construct subgraphs of different topological spaces (Rong et al. 2019). Each member in the ensemble model generates individual embedding from different topology spaces and has a powerful capacity to resist noise perturbations in graph data (Tang et al. 2021). MVE (Qu et al. 2017) promotes the collaboration of different views and lets them vote for robust representations with an attention mechanism. During the voting process in MVE, the attention mechanism enables each node to focus on the most important views. MGAT (Tao et al. 2020) utilizes an attention-based architecture to learn node representations from different multi-view. To collaboratively integrate multiple types of relationships in different views, MGAT aggregates the view-wise node representations via view-focused attention (Xie et al. 2020).

6.4 Spatio-temporal attention

In the real world, some graph-structured data often show dynamic properties with continuously evolving network nodes and topology over time, named dynamic graphs (Kazemi et al. 2020). Compared to static graph learning, learning representations on dynamic graphs is challenging due to the temporal dependencies over time. For dynamic graph representation learning, we usually divide the dynamic graph into different snapshots, according to different time windows, as shown in Fig. 15. Then, the dynamic graph representation learning learns low-dimensional node representations among a series of graph snapshots over the time step.

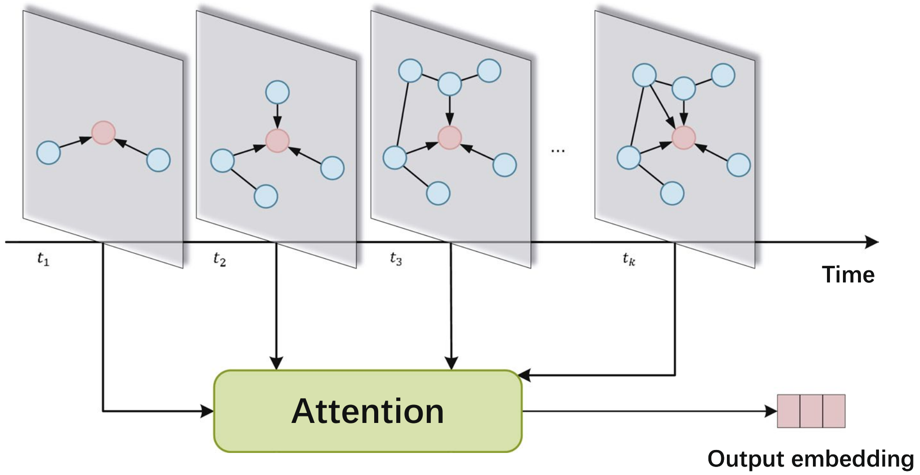


Fig. 15 Spatio-Temporal Attention in Dynamic Graph. Fusing messages from different time slices with attention mechanisms

DySAT (Sankar et al. 2018) learns node representations that capture both structural properties and temporal evolutionary patterns. In detail, DySAT generates node representations by jointly applying self-attention layers along two dimensions including structural neighborhood and temporal dynamics (Sankar et al. 2018). TemporalGAT (Fathy and Li 2020) employs the self-attention mechanism and structural neighborhoods over temporal dynamics with a temporal convolutional network. Considering the neighborhoods in different snapshots, TemporalGAT learns dynamic node representation via self-attention strategy without violating the ordering of the graph snapshots (Fathy and Li 2020).

For dynamic graph representation learning, GAEN (Shi et al. 2021a) provides an evolving graph attention network across different time points. In addition, GAEN allows attention weights to share and evolve across all temporal networks based on their respective topology discrepancies (Shi et al. 2021a). From the perspective of micro-and macro-dynamics, MMDNE (Lu et al. 2019) designs a temporal attention point process to capture structural and temporal properties at a fine-grained level. The micro-dynamics introduce the formation process of network structures in detail, while the macro-dynamics indicate the evolution pattern of the network scale (Lu et al. 2019). Considering continuous time in dynamic graphs, TGAT (Xu et al. 2020) presents a time-aware graph attention network to aggregate temporal-topological neighborhood features for inductive representation learning on temporal graphs. To handle continuous time, TGAT proposes a theoretically-grounded functional time encoding with a self-attention mechanism (Xu et al. 2020). Spatio-temporal attention has been successful in many application scenarios that can be modeled as dynamic graphs (Xu et al. , 2022; Yan et al. , 2018; Zheng et al. , 2020; Guo et al. , 2019; Fang et al. , 2020). In CV, spatio-temporal attention is applied to modeling a dynamic graph from a series of images at different times, such as trajectory prediction (Xu et al. 2022), and action recognition (Yan et al. 2018). Traffic prediction, including traffic flow forecasting (Zheng et al. 2020; Guo et al. 2019) and travel time estimation (Fang et al. 2020), is the most common application scenario of spatio-temporal attention because of the large amount of spatio-temporal traffic network data in our daily life.

6.5 Time series attention

A huge volume of data is generated overtime associated with various real-world systems. These data sets are often indexed by time, space, or both requiring appropriate approaches to analyze the data (Silva et al. 2021). Time series analysis usually works on common tasks, including forecasting, anomaly detection, and classification (Wen et al. 2022). If there are some association relationships between the time-series data that can be mined, graph neural networks can also play a role in time-series analysis. Numerous sensors have been deployed in different geospatial locations to continuously and cooperatively monitor the surrounding environment, such as climate science (Zhang et al. 2021c). As shown in Fig. 16, these sensors generate multiple geo-sensory time series, with spatial correlations between their readings (Liang et al. 2018).

To estimate the latent sensor graph structure and leverage the structure together with nearby observations, RainDrop (Zhang et al. 2021c) embeds irregularly sampled and multivariate time series while also learning the dynamics of sensors purely from observational data to predict misaligned readouts. For capturing time-varying dependencies among sensors, RainDrop can be interpreted as a graph neural network that sends messages over graphs with a self-attention mechanism (Zhang et al. 2021c). To capture the relationships between different time series explicitly, MTAD-GAT (Zhao et al. 2020) considers each

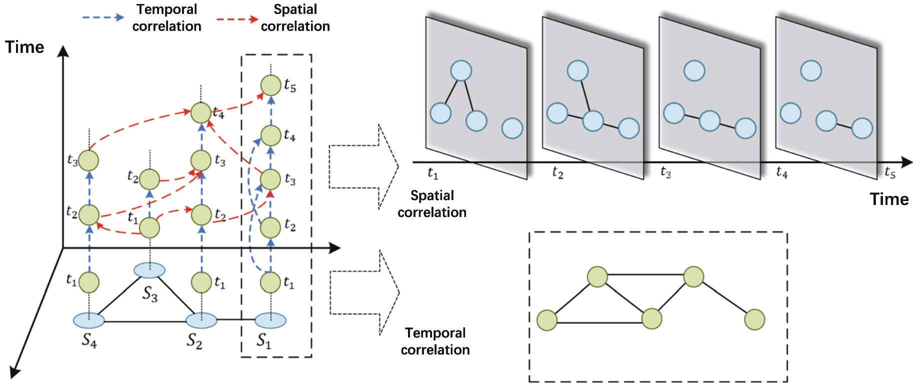


Fig. 16 Graph Attention in Time Series. Constructing graphs from a temporal and spatial perspective, modeling time series data with graphs

univariate time series as an individual feature and includes two graph attention layers in parallel to learn the complex dependencies of multivariate time series in both temporal and feature dimensions (Zhao et al. 2020). Two parallel graph attention layers in MTAD-GAT learn the relationships between different time series and timestamps dynamically (Zhao et al. 2020). GBikes (He and Shin 2020) proposes a data-driven Spatio-temporal Graph attention convolutional neural network (GACNN) for Bike station-level flow prediction. Based on the data-driven designs, GACNN predicts the fine-grained bike flows to/from each station, with attention mechanisms capturing and differentiating station-to-station correlations (He and Shin 2020).

7 Graph transformers

Taking into account whether the model uses the GNN layer to obtain adjacency information, we divide graph transformers into two sub-categories, i.e., standard Transformers and GNN Transformers. We enumerate the representative methods of standard Transformers and GNN Transformers in Table 3.

7.1 Standard transformers

Graph Transformers propagate node representations among all nodes in the graph regardless of whether two nodes are directly connected, while GAT focuses only on connected adjacent nodes. PAGAT (Chen et al. 2019) builds on longer-range dependencies in graph structure data and uses path features in molecular graphs to create a global attention layer.

As shown in Fig. 17, to utilize traditional Transformer architecture in the graph, it is necessary to define an effective position encoding with structural information of a graph, such as Laplacian encoding (Dwivedi and Bresson 2020), relation encoding (Cai and Lam 2020), centrality encoding, and edge encoding (Ying et al. 2021). GT (Dwivedi and Bresson 2020) proposes a generalization of transformer networks to homogeneous graphs of the arbitrary structure via Laplacian eigenvectors as positional

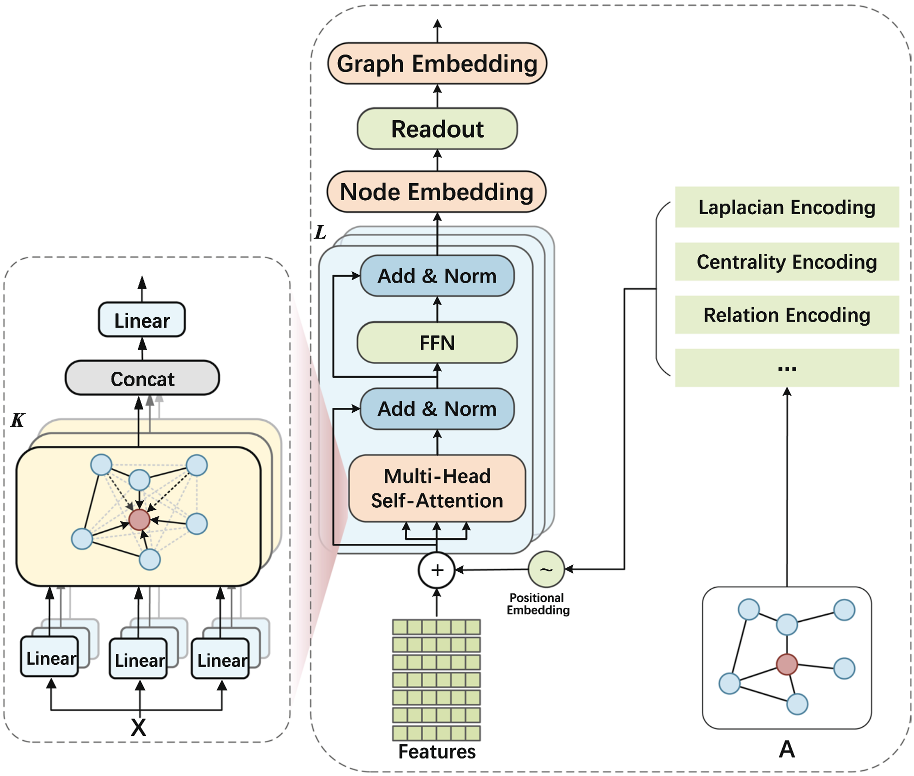


Fig. 17 The architecture of Graph Transformers. The attention mechanism in Graph Transformers can be seen as a fully connected GAT

features. By leveraging the full spectrum of the Laplacian, SAN (Kreuzer et al. 2021) can better detect similar sub-structures from their resonance with a fully-connected Transformer. Based on a batch of linkless subgraphs sampled from the original graph data, Graph-Bert (Zhang et al. 2020a) learns the representations of the target node with the extended graph-transformer layers effectively. Graphormer (Ying et al. 2021) is also built upon the standard Transformer architecture for graph representation learning tasks. For the graph-level tasks, the positional encodings in Graphormer are three novel graph structural encodings including centrality encoding, spatial encoding, and edge encoding (Ying et al. 2021).

Some research studies have explored graph representation learning using special mask mechanisms or higher-order Transformers. GTA (Seo et al. 2021) affords the advantages of both graph and sequence representations by encouraging the graph neural network characteristics of the transformer architecture. In addition, GTA takes the distance matrix of a molecular graph and atom-mapping matrix as mask self-attention and cross-attention. UniMP (Shi et al. 2021b) incorporates feature and label propagation at both training and inference time, taking feature embedding and masked label embedding as input information for propagation. Adopting the kernel attention approach to compute the pairwise weights, HOT (Kim et al. 2021b) generalizes

Transformers to any-order permutation invariant data involving sets, graphs, and hypergraphs.

7.2 GNN transformers

Only utilizing the self-attention mechanism to all nodes of the input graph, Standard Transformers usually ignore the information about the neighborhood structure of the nodes (Nguyen et al. 2019). To overcome this limitation, GNN Transformers consist of Transformer layers with GNN layers. The GNN layer in GNN Transformers is usually serial (before or after) with the Transformer layer, as shown in Fig. 18.

GraphFormers is a novel GNN Transformers architecture, where layerwise GNN components are nested alongside the Transformer blocks (Yang et al. 2021a). UGformer (Nguyen et al. 2019) presents two graph transformer variants leveraging the transformer on a set of sampled neighbors or all input nodes. HGT (Hu et al. 2020b) designs the heterogeneous mini-batch graph sampling algorithm for Web-scale heterogeneous graphs. To model heterogeneity, HGT designs node- and edge-type dependent parameters to characterize the heterogeneous attention over each edge, and maintain dedicated representations for different types of nodes and edges. To deliver a class of more expressive encoders of molecules, GROVER (Rong et al. 2020) introduces the message-passing mechanism into the Transformer architecture.

In an end-to-end fashion, GTN (Yun et al. 2019) learns to transform a heterogeneous input graph into useful meta-path graphs for each task and learns node representation on the graphs. The Transformer layer in GTN learns a soft selection of edge types and composite relations for generating useful multi-hop connections so-called meta-paths (Yun et al. 2019). To tackle the limitation of existing graph pooling methods, GMT (Baek et al. 2021) formulates the graph pooling problem as a multiset encoding problem with auxiliary information about the graph structure.

The Transformer model on graphs is more suitable for molecular graph tasks, molecular graph classification (Nguyen et al. 2019), molecular property prediction (Chen et al. 2019), and retrosynthesis tasks (Seo et al. 2021). Recently, graph transformers also stand out in

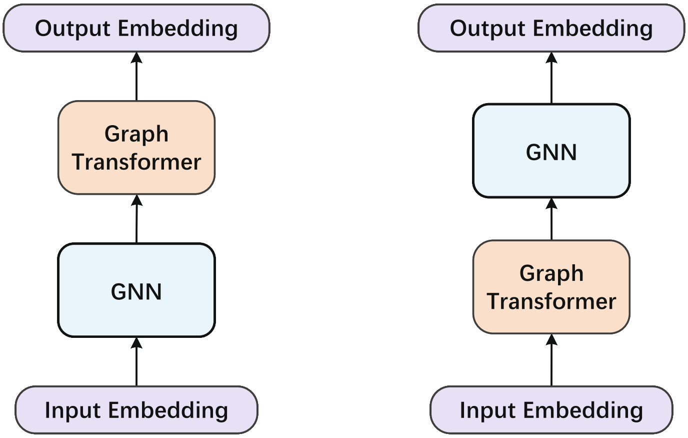


Fig. 18 The architecture of GNN Transformers. They can be seen as a combination of Graph Transformer and GNN

other application scenarios, such as NLP (Cai and Lam 2020; Koncel-Kedziorski et al. 2019), recommender systems (Xia et al. 2021b), knowledge graphs (Ahmad et al. 2021), and brain connections (Kim et al. 2021a).

8 Comparison and discussion

In this section, a model characteristics table is provided for a more comprehensive comparison. Each kind of model is characterized from nine perspectives, including local attention, high-order, global attention, multi-space, path-aware, edge-types, topology, times, and prior knowledge. In Table 4, ✓ indicates that most models in the subclass have the corresponding characteristic. “local attention”, “high-order” and “global attention” correspond to the information from neighborhoods that different models can obtain. “Multi-space” refers to the methods that adaptively select features from different feature spaces, resulting in a more robust latent representation. By considering the topological information of the graph, such as path, edge types, and topology, the model can learn a better representation that is more helpful for downstream tasks. These models are characterized by “path-aware”, “edge-types” and “topology”; Time is also an important property when learning the representation of nodes, especially in dynamic graphs and time series. This kind of model is characterized by “time”; “prior knowledge” refers to the models that add prior knowledge in the process of graph representation learning.

GRANs introduce RNN into graph representation learning, but at the same time, it is also limited by the inherent constraints of RNN (Liao et al. 2019). The original RNN architecture can only capture long-term dependencies on the ordered sequence, i.e. the output from one step will be used as input to the next step (Li et al. 2016). However, the neighbors of the central node are disordered in graphs (Hamilton et al. 2017). Although some methods deal with this problem by defining the order of nodes in advance (Hamilton et al. 2017), different arrangement orders may result in different performances. On the other hand, GRU-Attention and LSTM-Attention are both based on local attention, which focuses on immediate neighbors. It is difficult to obtain information from distant neighbors. To learn long-range patterns in graphs, GeniePath (Liu et al. 2019b) introduces skip connections. Some models in GRU-Attention take into account the edge type of node pair (Liao et al. 2019; Ruiz et al. 2020), while LSTM-Attention models prefer path-based random walks (Lee et al. 2018; Liu et al. 2019b).

Compared with GRANs, GATs can be calculated in parallel with the attention mechanism, because there is no need for sequential calculation. The attention mechanisms of GATs allow for dealing with variable-sized inputs, focusing on the most relevant parts of the input (Veličković et al. 2018). Almost all models in intra-layer GATs aggregate and update the representation of nodes in defined local neighborhoods with local attention. GATs with local attention (Veličković et al. 2018; Wang et al. 2019a; Zhang and Xie 2020; Kim and Oh 2021; Brody et al. 2021) enable specifying different weights to different nodes in the local neighborhood. However, GAT (Veličković et al. 2018) also suffers from over-fitting and over-smoothing (Wang et al. 2019a). To address the above weaknesses, C-GAT improves GAT via margin-based constraints on attention during training (Wang et al. 2019a). To make a clear understanding of the discriminative capacities of GAT, CPA presents a theoretical analysis of the representational properties of the attention-based GNNs (Zhang and Xie 2020). Considering the attribute homophily rate, DMP specifies every attribute propagation weight on each edge in graphs with heterophily (Yang

Table 4 Characteristics of different models in subclasses from nine perspectives

Stages	Subclasses	Local	High order	Global	Multi space	Path	Edge types	Topology	Time	Prior
GRANs	GRU- Attention	✓					✓			
	LSTM- Attention	✓				✓				
Intra-Layer GATs	Neighbor Attention	✓								
	High-Order Attention	✓	✓			✓				✓
	Relation- Aware Attention	✓					✓			✓
	Hierarchical Attention	✓	✓		✓	✓	✓			✓
	Attention Sampling/Pooling	✓						✓		
	Hyper- Attention	✓	✓				✓			✓
Inter-Layer GATs	Multi-Level Attention	✓	✓		✓					✓
	Multi-Channel Attention				✓					✓
	Multi-View Attention				✓			✓		
	Spatio-Temporal Attention				✓				✓	✓
	Time Series Attention				✓			✓	✓	
Graph Transformers	Standard Transformers		✓	✓		✓	✓	✓		✓
	GNN Transformers	✓	✓	✓		✓	✓	✓		✓

et al. 2021b). To enable embeddings with much smaller distortion beyond Euclidean space, hyperbolic GATs (Chami et al. , 2019; Zhang and Gao , 2021; Zhang et al. , 2021d) learn embeddings that preserve hierarchical structure in hyperbolic space. Unfortunately, It is difficult for these methods to directly obtain information from distant neighbors. To alleviate this limitation, GATs with high-order attention attempt to obtain information from distant neighbors through path-aware strategies (Yang et al. 2019; Cao et al. 2020; Yang et al. 2021d). To preserve the hierarchy in the graph topology, hierarchical attention defines two or more levels of attention mechanisms. Low-level attention is based on the node, while high-level attention can be based on paths (Yang et al. 2021e), relations (Zhang et al. 2020c), or groups (Lin et al. 2022). In such a case, hierarchical attention can learn specific hierarchical features of the graph. For graph classification, attention-based pooling (Huang et al. 2019a; Lee et al. 2019a) can learn hierarchical representations based on the attention mechanism. To apply GATs to complex graphs with multiple edge types, models with relation-aware attention usually define different relationships based on edge types (Huang et al. 2019b; Hong et al. 2020; Qin et al. 2021b). While many complex graphs consist of relationships beyond pair-wise interactions, i.e., hyperedge that connects more than two nodes (Zhang et al. 2020b). Models with hyper-attention are designed for hypergraphs to extract patterns among higher-order interactions (Zhang et al. 2021a).

Different from intra-layer GATs, inter-layer GATs extract useful hidden representations from different feature spaces via feature fusion attention, not just local neighborhoods. However, the performance of attention-based GNNs decreases when going deeper. Several recent studies attribute this performance deterioration to the entanglement of representation transformation and propagation in GNNs (Liu et al. 2020). After decoupling these two operations, deeper GNNs can be used to learn node representations from larger receptive fields (Liu et al. 2020; Zhang et al. 2022c). Then, multi-level attention adaptively fuses the representation of different-order neighbors to obtain high-order information. Similarly, multichannel attention can distinguish signals with different frequencies during message passing (Bo et al. 2021). To obtain the representations from different feature spaces, models with multi-view attention first construct several graphs with different topological structures based on the original graph and then fuse the representations of different views to obtain the final representation (Yang et al. 2021e; Tao et al. 2020). However, the above GATs ignore the graph with the time attribute. Spatio-Temporal attention and time-series attention can handle data with time attributes, such as dynamic graphs (Sankar et al. 2018; Fathy and Li 2020; Xu et al. 2020) and time series (Shukla and Marlin 2021; Zhang et al. 2021c). Especially, time-series attention should first construct the graph structure from the time-series data.

With the advent of Graph Transformers, GNNs do not need to obtain distant messages by stacking network layers. Graph Transformers can directly propagate the information among the nodes in the whole graph with global attention. Standard Transformers in graphs generate the position encoding with path-aware (Ying et al. 2021), edge-types (Dwivedi and Bresson 2020), or additional attributes (Kreuzer et al. 2021; Zhang et al. 2020a). This kind of models are outstanding on the small molecular graph. However, Standard Transformers do not consider the original graph structure during training. To make up for this shortcoming, GNN Transformers jointly train the GNN layers and Transformers layers (Zhang et al. 2020a). This combination of models can effectively compensate for the shortcomings of the above single model. Further, we can capture both local and global information through a combination of local and global attention, which is difficult to achieve with a single model.

In addition, it is also a good choice for the combination model of different attention, which can be of the same type Choi (2022); Kim and Oh (2021) or different types (Zhang et al. 2020c; Yang et al. 2021e; Zheng et al. 2020). This combination may make up for the deficiency of a single model, thus improving the expression ability of the model. For example, SuperGAT takes advantage of different attention operations to learn label agreement and edge presence (Kim and Oh 2021). Typically, methods with hierarchical attention tend to include two and more attention mechanisms, which can capture different levels of semantic information. The hierarchical attention mechanism utilizes the neighborhood information of an entity more effectively from different levels (Zhang et al. 2020c). For HAN in heterogeneous graphs, the node-level attention aims to learn the importance between a node and its neighbors along the meta-path, while the semantic-level attention can learn the importance of different meta-paths (Wang et al. 2019d). More complexly, GMAN (Zheng et al. 2020) proposes a graph multi-attention network for long-term traffic prediction, which includes spatial attention, temporal attention, and gated fusion. However, the simultaneous use of multiple attention mechanisms in the same model will also significantly increase the computational complexity. In short, it is difficult to adapt any model to all scenarios. We hope to help researchers better design their attention-based GNNs.

To the best of our knowledge, it is difficult to quantitatively compare and analyze the complexity of different models, especially when different model structures and different attention mechanisms are involved. Therefore, we provide a general analysis of three basic attention-based GNNs. The time complexity of GRANs is high due to their recursive nature. The updating of node representations needs to consider the representations of their neighboring nodes at each time step, and this process requires multiple iterations over time steps. GATs have a high time complexity because they leverage self-attention mechanisms to compute attention weights between nodes. This node-to-node attentional computation increases the space-time complexity. The attention weights can be computed in parallel, which can help to reduce the time complexity but increases the space complexity at the same time. In addition, the scale of graphs continues to grow exponentially over time in real life. This will be a huge challenge for GATs, as well as for GRANs and Graph Transformers. Graph Transformers, based on the Transformer model, involve self-attention and multi-head self-attention in neural network layers. The attention mechanism in Graph Transformers can be seen as a fully connected GAT, and we can imagine that its complexity is usually much higher than that of GAT. The actual time complexity depends on factors such as the specific model architecture, graph size, sparsity, and available computational resources.

9 Open issues and future directions

9.1 Scalability

Scalability is a major challenge for the application of attention-based GNNs in practical scenarios. Although there are some open source medium or large-scale graph datasets in OGB (Hu et al. 2020a), most of the existing studies focus on testing the new developed attention-based GNNs on small graphs (e.g., some citation network datasets with only a few thousand nodes such as Cora and Citeseer (Welling and Kipf 2016)), while ignoring the huge, complex and noisy networks in practical applications. Large-scale graph data has been a huge challenge for attention-based GNNs. The most fundamental reason for this phenomenon in attention-based GNNs is the high computational complexity of the attention mechanism. Attention-based GNNs are hard to implement in large graphs, especially for Graph Transformers. Up to now, although some approaches have attempted to improve model efficiency through sampling and subgraphs (Zhang et al. 2020a; Nguyen et al. 2019; Hu et al. 2020b), they still can not efficiently process large graphs.

9.2 Interpretability

Though attention-based GNNs achieve promising performance on various tasks, a clear understanding of their discriminative power is superficial (Zhang and Xie 2020). Without understanding the relationships behind the predictions, these models can not be understood and fully trusted (Yuan et al. 2021a). Recent works (Xu et al. 2018a; Maron et al. 2019) attempt to explore the interpretability of graph neural networks and theoretically analyze the expressive power of GNNs. To improve the performance of attention-based GNNs, CPA (Zhang and Xie 2020) improves the attention mechanism in GNNs via cardinality preservation and presents a theoretical analysis of the representational properties of attention-based GNNs. HAGERec proposes a hierarchical attention graph convolutional network to explore users' potential preferences from the high-order connectivity structure (Yang and Dong 2020). Considering the topology of the graph, some researches are carried out from the perspective of motifs (Peng et al. 2020) and subgraph (Yuan et al. 2021a). However, there is still a lot of room for the interpretability of attention-based GNNs (Yuan et al. 2020).

9.3 Deeper models

Like traditional GNNs, most attention-based GNNs aggregate messages from local neighbors iteratively, ignoring messages from the distant neighborhood. When the downstream task depends on long-range interaction, GNNs fail to propagate messages originating from distant neighborhoods and perform poorly. Although they can obtain distant messages by stacking the neural network layers, this also brings about the over-smoothing (Li et al. 2018), i.e., node representations become indistinguishable. To improve expressive capability and alleviate the over-smoothing of attention-based GNNs, some researchers focus on the topological properties of graphs (Zhang and Xie 2020; Yang et al. 2021b). Focus on the architecture of GNNs, decoupling representation transformation and message propagation enable deeper GNNs with attention to learning graph node representations from larger receptive fields (Liu et al. 2020; Zhang et al. 2022c). As the number of layers increases, the receptive field of a node grows exponentially. This causes over-squashing (Alon and Yahav 2020), where information from the exponentially-growing receptive field is compressed into fixed-length node vectors. Compared with GCNs, GATs can effectively alleviate the over-squashing phenomenon by introducing an attention mechanism (Alon and Yahav 2020). However, when stacking too many layers, GATs also suffer from higher complexity. Instead, Graph Transformers (Zhang et al. 2020a; Nguyen et al. 2019) obtain global information in a network layer, which is a breakthrough to alleviate the problem of over-squashing. In recent years, many researchers explore the above problems from the perspective of higher-order structures (Kim et al. , 2021b; Battiston et al. , 2021), such as hypergraphs and simplicial complexes.

9.4 Complex graph

Graph-structured data is ubiquitous in our daily life, such as social networks, citation networks, and collaboration networks. In the real-world, however, graphs can be both structurally large and complex (Lee et al. 2019b). In addition to the above graph-structured data often used in scientific research, there are many more complex graph scenarios with a large amount of semantic information used in industry, such as transportation networks (Zheng et al. 2020), knowledge graphs (Wang et al. 2019c), and chemical molecule graphs (Chen et al. 2019). These real-world networks are usually modeled as homogeneous graphs (Wang et al. 2022), relation-aware graphs (Qin et al. 2021b), or dynamic graphs (Zhu et al. 2022). However, in the case of heterogeneous graphs, attention-based GNNs require specific modifications to effectively integrate and combine the semantic information across nodes, edges, and graphs. For dynamic graphs with time attributes, the graph structure evolves with time, which increases the difficulty of attention-based GNNs training. Therefore, how to design a more universal attention-based GNN, friendly to graphs with different complex attributes, is still an open problem.

9.5 Novel applications

Attention-based GNNs are widely used in social networks, natural language processing, recommendation systems, and traffic forecasting. In CV that seems to be unrelated to the graph, attention-based GNNs have also begun to be widely applied by generating graphs. Attention-based GNNs are widely used in social networks (Su et al. 2022), natural language processing (Wu et al. 2021b), recommendation systems (Wu et al. 2020a), traffic forecasting (Jiang and Luo 2021), and multimodal (Ektefaie et al. 2023). In CV that seems to be unrelated to the graph, attention-based GNNs have also begun to be widely applied by generating graphs (Xu et al. 2022). Similarly, we can model segment-wise speaker embeddings (SSEs) as nodes of a graph (Jung et al. 2021b). Further, GNNs could model interactions both within and across different data types in multimodal data, including image, video, language, and knowledge graph. Once complex relations between modalities can be built into a network structure, GNNs provide a powerful and flexible strategy to leverage interdependencies in multimodal datasets (Ektefaie et al. 2023). Combinatorial optimization is a canonical NP-hard problem in traditional operations research. GNN as a scalable general-purpose solver is adopted to approximately solve combinatorial optimization problems (Schuetz et al. 2022). In the field of biochemistry, GNNs also show unique advantages, especially in generating molecular graphs (Stärk et al. 2021). It can be seen that attention-based GNNs have good application prospects. There are still many new fields waiting for us to explore, especially in scenes that can be modeled with graphs.

10 Conclusion

Over the past decade, the attention mechanism has achieved impressive performance in NLP and CV. Attention-based GNNs allow us to adaptively aggregate and update representations from the different neighborhoods with local or global attention, even feature fusion attention. In this survey, we provide a comprehensive review of the most recent research efforts on attention-based GNNs. We first give some basic definitions of attention-based GNNs including graph, graph neural networks, and attention mechanisms. Then, we propose a novel two-level taxonomy for attention-based GNNs from the perspective of development history and architectural perspectives. To be specific, we classify existing attention-based GNNs into three stages and then organize them into several sub-categories in each stage according to the model architecture. For each sub-category, we briefly clarify the main characteristics, detail the attention strategies adopted by the representative models, and discuss their advantages and limitations. Furthermore, we outline some open issues and promising future research directions to advance this field. Finally, we intend to share the relevant latest papers about attention-based GNNs at the open resources. We hope this survey will provide a succinct introduction to attention-based GNNs and shed some light on future developments.

Appendix

For ease of reading, we summarize the abbreviations of the methods and their corresponding full names in Table 5.

Table 5 Abbreviations of the methods and their corresponding full names

Abbreviation	Full Name
SpectralCNN	Spectral Convolutional Neural Networks
ChebNet	Graph CNN with Chebyshev Filter
GCN	Graph Convolutional Networks
MPNN	Message-Passing Neural Networks
GraphNet	Graph Networks
GAT	Graph Attention Networks
C-GAT	Constrained Graph Attention Networks
SuperGAT	Self-supervised Graph Attention Network
GRU	Gated Recurrent Units
GGNN	Gated Graph Sequence Neural Networks
GRNN	Graph Recurrent Neural Networks
GSP	graph signal processing
GaAN	Gated Attention Networks
GRAN	Graph Recurrent Attention Networks
GraphSAGE	Graph SAmple and aggreGatE
LSTM	Long Short-Term Memory
JK-Net	Jumping Knowledge Networks
GAM	Graph Attention Model
DMP	Diverse Message Passing
PPRGAT	Personalized PageRank GAT
GANet	Graph Attention Networks
CAT	Conjoint Attention Networks
MSNA	MultiSpace Neighbor Attention
CPA	Cardinality Preserved Attention
AGNN	Attention-based Graph Neural Network
HTNE	Hawkes process based Temporal Network Embedding
HAT	Hyperbolic Graph Attention Network
HGCN	Hyperbolic Graph Convolutional Neural Network

Table 5 (continued)

Abbreviation	Full Name
Hype-HAN	Hy-perbolic Hierarchical Attention Network
SPAGAN	Shortest Path Graph Attention Network
PaGNN	Path-aware Graph Neural Network
ADSF	Adaptive Structural Fingerprint
MAGNA	Multi-hop Attention Graph Neural Networks
T-GAP	Temporal Knowledge Graph with Attention Propagation
SiGAT	Signed Graph Attention Networks
SNEA	Signed Network Embedding via Graph Attention
RGAT	Relational Graph Attention Networks
HetSANN	Heterogeneous Graph Structural Attention Neural Network
WRGAT	Weighted Relational GAT
EAGCN	Edge Attention-based Multi-Relational GCN
TALP	Type-Aware Anchor Link Prediction
KGAT	Knowledge Graph Attention Network
GATE	Graph Attention Transformer Encoder
RelGNN	Relation-aware GNN
CGAT	Channel-aware Graph Attention Network
PSHGAN	Meta-path and meta-structure integrated heterogeneous graph neural network through attention mechanisms
PRML	Path-based Proximity Ranking Metric Dual-Level Attention Learning
RGHAT	Relational Graph neural network with Hierarchical ATtention
LAN	Logic Attention Network
GraphHAM	Graph embedding model with Hierarchical Attentive Memberships
GAW	Graph Attention model with random Walk
NLGAT	Non-Local GAT
DiffPool	Differentiable Graph Pooling
SAGPool	Self-Attention Graph Pooling
AttPool	graph pooling module based on an attention based mechanism
ChebyGIN	Combining GIN with ChebyNet
DAGNN	Deep Adaptive Graph Neural Network
TDGNN	Tree Decomposed Graph Neural Network
FAGCN	Frequency Adaptation Graph Convolutional Networks
ACM	Adaptive Channel Mixing
AM-GCN	Adaptive Multi-channel GCN
UAG	Uncertainty-aware Attention Graph Neural Network
MV-GCN	Multi-View Graph Convolutional Netowork
GENet	Graph Ensemble Network
MVE	Multi-View Network Representation Learning
MGAT	Multi-view Graph Attention Networks
DySAT	Dynamic Self-Attention Network
TemporalGAT	Temporal-GAT
GAEN	Graph Attention Evolving Networks
MMDNE	Network Embedding with Micro- and Macro-Dynamics
TGAT	Temporal Graph Attention
MTAD-GAT	Multivariate Time-series Anomaly Detection via Graph Attention Network

Table 5 (continued)

Abbreviation	Full Name
GACNN	Graph Attention Convolutional Neural Network
Hyper-SAGNN	Self-Attention based Graph Neural Network for Hypergraphs
HHGR	Hierarchical Hypergraph Learning Framework for Group Recommendation
HyperTeNet	Hypergraph and Transformer-based Neural Network
Hyper-GAT	Hypergraph-GAT
PAGAT	Path-Augmented Graph Transformer Networks
GT	Graph Transformer
SAN	Spectral Attention Network
GTA	Graph Truncated Attention
UniMP	Unified Message Passaging Model
HOT	Higher-Order Transformer
GraphFormers	GNN-nested Transformers
UGformer	Universal Graph Transformer
HGT	Heterogeneous Graph Transformer
GROVER	Graph Representation frOm self-superVised mEssage passing tRansformer
GTN	Graph Transformer Networks
GMT	Graph Multi-set Transformer
OGB	Open Graph Benchmark

Acknowledgements This work was supported by the National Natural Science Foundation of China (No.61876186) and the Xuzhou Science and Technology Project (No.KC21300).

References

Abu-El-Haija S, Perozzi B, Al-Rfou R et al (2018) Watch your step: learning node embeddings via graph attention. Adv Neural Inf Processing Syst. https://doi.org/10.48550/arXiv.1710.09599

Ahmad WU, Peng N, Chang KW (2021) Gate: graph attention transformer encoder for cross-lingual relation and event extraction. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 12,462–12,470

Alon U, Yahav E (2020) On the bottleneck of graph neural networks and its practical implications. Mach Learn 89:5–35

Baek J, Kang M, Hwang SJ (2021) Accurate learning of graph representations with graph multiset pooling. In: The Ninth International Conference on Learning Representations, The International Conference on Learning Representations (ICLR)

Bai S, Zhang F, Torr PH (2021) Hypergraph convolution and hypergraph attention. Pattern Recognit 110(107):637

Battaglia PW, Hamrick JB, Bapst V, et al (2018) Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261

Battiston F, Amico E, Barrat A et al (2021) The physics of higher-order interactions in complex systems. Nat Phys 17(10):1093–1098

Bo D, Wang X, Shi C, et al (2021) Beyond low-frequency information in graph convolutional networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 3950–3957

Brauwers G, Frasincar F (2021) A general survey on attention mechanisms in deep learning. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2021.3126456

Brody S, Alon U, Yahav E (2021) How attentive are graph attention networks? In: International Conference on Learning Representations

Bruna J, Zaremba W, Szlam A, et al (2014) Spectral networks and locally connected networks on graphs. In: International Conference on Learning Representations (ICLR2014), CBLS, April 2014, pp http–openreview

Busbridge D, Sherburn D, Cavallo P, et al (2019) Relational graph attention networks. arXiv preprint arXiv:1904.05811

Cai D, Lam W (2020) Graph transformer for graph-to-sequence learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 7464–7471

Cao M, Ma X, Zhu K, et al (2020) Heterogeneous information network embedding with convolutional graph attention networks. In: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, pp 1–8

Cen Y, Zou X, Zhang J, et al (2019) Representation learning for attributed multiplex heterogeneous network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1358–1368

Chami I, Ying Z, Ré C, et al (2019) Hyperbolic graph convolutional neural networks. Advances in neural information processing systems 32

Chaudhari S, Mithal V, Polatkan G et al (2021) An attentive survey of attention models. ACM Trans Intell Syst Technol (TIST) 12(5):1–32

Chen B, Barzilay R, Jaakkola T (2019) Path-augmented graph transformer network. Mach Learn. https://doi.org/10.48550/arXiv.1905.12712

Cheng R, Li Q (2021) Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 55–62

Choi E, Bahadori MT, Song L, et al (2017) Gram: graph-based attention model for healthcare representation learning. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 787–795

Choi J (2022) Personalized pagerank graph attention networks. ICASSP 2022–2022 IEEE International Conference on Acoustics. Speech and Signal Processing (ICASSP), IEEE, pp 3578–3582

Cini A, Marisca I, Bianchi FM, et al (2022) Scalable spatiotemporal graph neural networks. arXiv preprint arXiv:2209.06520

Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. Adv Neural Inf Processing Syst. https://doi.org/10.48550/arXiv.1606.09375

Dwivedi VP, Bresson X (2020) A generalization of transformer networks to graphs. Mach Learn. https://doi.org/10.48550/arXiv.2012.09699

Ektefaie Y, Dasoulas G, Noori A et al (2023) Multimodal learning with graphs. Nat Mach Intell. https://doi.org/10.1038/s42256-023-00624-6

Fang X, Huang J, Wang F, et al (2020) Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 2697–2705

Fathy A, Li K (2020) Temporalgat: attention-based dynamic graph representation learning. Pacific-Asia conference on knowledge discovery and data mining. Springer, Berlin, pp 413–423

Feng B, Wang Y, Ding Y (2021) Uag: Uncertainty-aware attention graph neural network for defending adversarial attacks. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 7404–7412

Gao C, Wang X, He X, et al (2022) Graph neural networks for recommender system. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp 1623–1625

Gao H, Ji S (2019) Graph representation learning via hard and channel-wise attention networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 741–749

Georgousis S, Kenning MP, Xie X (2021) Graph deep learning: State of the art and challenges. IEEE Access 9:22

Gilmer J, Schoenholz SS, Riley PF, et al (2017) Neural message passing for quantum chemistry. In: International Conference on Machine Learning, PMLR, pp 1263–1272

Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 855–864

Gulcehre C, Denil M, Malinowski M et al (2018) Hyperbolic attention networks. Neural Evol Comput. https://doi.org/10.48550/arXiv.1805.09786

Guo MH, Xu TX, Liu JJ et al (2022) Attention mechanisms in computer vision: a survey. Comput Vis Media 3:1–38

Guo S, Lin Y, Feng N, et al (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 922–929

Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. Adv Neural Inf Processing Syst. https://doi.org/10.1093/bioinformatics/btad135

He S, Shin KG (2020) Towards fine-grained flow forecasting: a graph attention approach for bike sharing systems. Proc Web Conf 2020:88–98

He T, Ong YS, Bai L (2021) Learning conjoint attentions for graph neural nets. Adv Neural Inf Proc Syst 34:2641–2653

Hong H, Guo H, Lin Y, et al (2020) An attention-based graph neural network for heterogeneous structural learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 4132–4139

Hu W, Fey M, Zitnik M et al (2020) Open graph benchmark: datasets for machine learning on graphs. Adv Neural Inf Processing Syst 33:22

Hu Z, Dong Y, Wang K et al (2020) Heterogeneous graph transformer. Proc Web Conf 2020:2704–2710

Huang B, Carley KM (2019) Syntax-aware aspect level sentiment classification with graph attention networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp 5469–5477

Huang J, Li Z, Li N et al (2019a) Attpool: Towards hierarchical feature representation in graph convolutional networks via attention mechanism. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 6480–6489

Huang J, Shen H, Hou L et al (2019) Signed graph attention networks. Int Conf Artif Neural Netw. Springer, Berlin, pp 566–577

Jiang W, Luo J (2021) Graph neural network for traffic forecasting: A survey. arXiv preprint arXiv:2101.11174

Jung J, Jung J, Kang U (2021a) Learning to walk across time for interpretable temporal knowledge graph completion. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp 786–795

Jw Jung, Heo HS, Yu HJ et al (2021) Graph attention networks for speaker verification. ICASSP 2021–2021 IEEE International Conference on Acoustics. Speech and Signal Processing (ICASSP), IEEE, pp 6149–6153

Kazemi SM, Goel R, Jain K et al (2020) Representation learning for dynamic graphs: a survey. J Mach Learn Res 21(70):1–73

Kim BH, Ye JC, Kim JJ (2021) Learning dynamic graph representation of brain connectome with Spatio-temporal attention. Adv Neural Inf Proc Syst 34:4314–4327

Kim D, Oh AH (2021) How to find your friendly neighborhood: Graph attention design with self-supervision. In: The Ninth International Conference on Learning Representations (ICLR 2021), International Conference on Learning Representations (ICLR 2021)

Kim J, Oh S, Hong S (2021) Transformers generalize deepsets and can be extended to graphs & hypergraphs. Adv Neural Inf Proc Syst 34:28,016-28,028

Kim J, Yoon S, Kim D et al (2021c) Structured co-reference graph attention for video-grounded dialogue. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 1789–1797

Klicpera J, Bojchevski A, Günnemann S (2018) Predict then propagate: graph neural networks meet personalized Pagerank. Comput Sci. https://doi.org/10.48550/arXiv.1810.05997

Klicpera J, Weißenberger S, Günnemann S (2019) Diffusion improves graph learning. arXiv preprint arXiv:1911.05485

Knyazev B, Taylor GW, Amer M (2019) Understanding attention and generalization in graph neural networks. Adv Neural Inf Proc Syst. https://doi.org/10.48550/arXiv.1905.02850

Koncel-Kedziorski R, Bekal D, Luan Y, et al (2019) Text generation from knowledge graphs with graph transformers. In: 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (ACL), pp 2284–2293

Kreuzer D, Beaini D, Hamilton W et al (2021) Rethinking graph transformers with spectral attention. Adv Neural Inf Proc Syst 34:21,618-21,629

Lee J, Lee I, Kang J (2019) Self-attention graph pooling. International conference on machine learning. PMLR, New York, pp 3734–3743

Lee JB, Rossi R, Kong X (2018) Graph classification using structural attention. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1666–1674

Lee JB, Rossi RA, Kim S et al (2019) Attention models in graphs: a survey. ACM Trans Knowl Dis Data (TKDD) 13(6):1–25

Li J, Liu X, Zong Z et al (2020a) Graph attention based proposal 3d convnets for action detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 4626–4633

Li L, Gan Z, Cheng Y et al (2019) Relation-aware graph attention network for visual question answering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 10,313–10,322

Li Q, Han Z, Wu XM (2018) Deeper insights into graph convolutional networks for semi-supervised learning. In: 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, AAAI press, pp 3538–3545

Li X, Shang Y, Cao Y et al (2020b) Type-aware anchor link prediction across heterogeneous networks based on graph attention network. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 147–155

Li Y, Zemel R, Brockschmidt M et al (2016) Gated graph sequence neural networks. In: Proceedings of ICLR'16

Li Y, Tian Y, Zhang J et al (2020c) Learning signed network embedding via graph attention. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 4772–4779

Liang Y, Ke S, Zhang J et al (2018) Geoman: Multi-level attention networks for geo-sensory time series prediction. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp 3428–3434

Liao R, Li Y, Song Y et al (2019) Efficient graph generation with graph recurrent attention networks. Adv Neural Inf Proc Syst. https://doi.org/10.1016/j.aiopen.2021.01.001

Lin L, Wang H (2020) Graph attention networks over edge content-based channels. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1819–1827

Lin L, Blaser E, Wang H (2022) Graph embedding with hierarchical attentive membership. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp 582–590

Lin T, Wang Y, Liu X et al (2021) A survey of transformers. arXiv preprint arXiv:2106.04554

Liu M, Gao H, Ji S (2020) Towards deeper graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 338–348

Liu M, Wang Z, Ji S (2021) Non-local graph neural networks. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.4108/eetel.v8i3.3461

Liu S, Chen Z, Liu H et al (2019a) User-video co-attention network for personalized micro-video recommendation. In: The World Wide Web Conference, pp 3020–3026

Liu Z, Chen C, Li L et al (2019b) Geniepath: Graph neural networks with adaptive receptive paths. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 4424–4431

Long Y, Wu M, Liu Y et al (2021) Graph contextualized attention network for predicting synthetic lethality in human cancers. Bioinformatics 37(16):2432–2440

Lu Y, Wang X, Shi C et al (2019) Temporal network embedding with micro-and macro-dynamics. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp 469–478

Luan S, Hua C, Lu Q et al (2021) Is heterophily a real nightmare for graph neural networks to do node classification? arXiv preprint arXiv:2109.05641

Lv Q, Ding M, Liu Q et al (2021) Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp 1150–1160

Ma N, Mazumder S, Wang H et al (2020) Entity-aware dependency-based deep graph attention network for comparative preference classification. In: Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2020)

Maron H, Ben-Hamu H, Serviansky H et al (2019) Provably powerful graph networks. Adv Neural Inf Proc Syst. https://doi.org/10.1038/s43246-022-00315-6

Mei G, Pan L, Liu S (2022) Heterogeneous graph embedding by aggregating meta-path and meta-structure through attention mechanism. Neurocomputing 468:276–285

Min E, Chen R, Bian Y et al (2022) Transformer for graphs: An overview from architecture perspective. arXiv preprint arXiv:2202.08455

Mou C, Zhang J, Wu Z (2021) Dynamic attentive graph learning for image restoration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 4328–4337

Nathani D, Chauhan J, Sharma C et al (2019) Learning attention-based embeddings for relation prediction in knowledge graphs. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp 4710–4723

Nguyen DQ, Nguyen TD, Phung D (2019) Universal graph transformer self-attention networks. arXiv preprint arXiv:1909.11855

Peng H, Li J, Gong Q et al (2020) Motif-matching based subgraph-level attentional convolutional network for graph classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 5387–5394

Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 701–710

Phuong M, Hutter M (2022) Formal algorithms for transformers. arXiv preprint arXiv:2207.09238

Qin L, Li Z, Che W et al (2021a) Co-gat: A co-interactive graph attention network for joint dialog act recognition and sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 13,709–13,717

Qin X, Sheikh N, Reinwald B et al (2021b) Relation-aware graph attention model with adaptive self-adversarial training. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 9368–9376

Qu M, Tang J, Shang J et al (2017) An attention-based collaboration framework for multi-view network representation learning. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp 1767–1776

Rong Y, Huang W, Xu T et al (2019) Dropedge: Towards deep graph convolutional networks on node classification. In: International Conference on Learning Representations

Rong Y, Bian Y, Xu T et al (2020) Self-supervised graph transformer on large-scale molecular data. Adv Neural Inf Proc Syst 33:12,559-12,571

Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326

Ruiz L, Gama F, Ribeiro A (2020) Gated graph recurrent neural networks. IEEE Tran Signal Proc 68:6303–6318

Sankar A, Wu Y, Gou L et al (2018) Dynamic graph representation learning via self-attention networks. arXiv preprint arXiv:1812.09430

Scarselli F, Gori M, Tsoi AC et al (2008) The graph neural network model. IEEE Trans Neural Netw 20(1):61–80

Schuetz MJ, Brubaker JK, Katzgraber HG (2022) Combinatorial optimization with physics-inspired graph neural networks. Nat Mach Intell 4(4):367–377

Seo SW, Song YY, Yang JY et al (2021) Gta: Graph truncated attention for retrosynthesis. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 531–539

Shang C, Liu Q, Chen KS et al (2018) Edge attention-based multi-relational graph convolutional networks. arXiv preprint arXiv:1802.04944

Shi M, Huang Y, Zhu X et al (2021a) Gaen: Graph attention evolving networks. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp 1541–1547

Shi Y, Huang Z, Feng S et al (2021b) Masked label prediction: Unified message passing model for semi-supervised classification. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, pp 1548–1554

Shukla SN, Marlin BM (2021) Multi-time attention networks for irregularly sampled time series. arXiv preprint arXiv:2101.10318

Silva VF, Silva ME, Ribeiro P et al (2021) Time series analysis via network science: concepts and algorithms. Wiley Interdiscip Rev 11(3):e1404

Stachenfeld K, Godwin J, Battaglia P (2020) Graph networks with spectral message passing. arXiv preprint arXiv:2101.00079

Stärk H, Beaini D, Corso G et al (2021) 3d infomax improves gnns for molecular property prediction. arXiv preprint arXiv:2110.04126

Su X, Xue S, Liu F et al (2022) A comprehensive survey on community detection with deep learning. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2021.3137396

Sun Q, Liu H, He J et al (2020) Dagc: Employing dual attention and graph convolution for point cloud based place recognition. In: Proceedings of the 2020 International Conference on Multimedia Retrieval, pp 224–232

Suresh S, Budde V, Neville J et al (2021) Breaking the limit of graph neural networks by improving the assortativity of graphs with local mixing patterns. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp 1541–1551

Tang H, Liang X, Wu B et al (2021) Graph ensemble networks for semi-supervised embedding learning. International Conference on Knowledge Science. Springer, Engineering and Management, pp 408–420

Tang J, Qu M, Wang M et al (2015) Line: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp 1067–1077

Tao Z, Wei Y, Wang X et al (2020) Mgat: multimodal graph attention network for recommendation. Inf Proc Manag 57(5):102,277

Tay Y, Dehghani M, Bahri D et al (2020) Efficient transformers: a survey. ACM Compu Surveys (CSUR). https://doi.org/10.48550/arXiv.2009.06732

Thekumparampil KK, Wang C, Oh S et al (2018) Attention-based graph neural network for semi-supervised learning. arXiv preprint arXiv:1803.03735

Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. Adv Neural Inf Proc Syst 30:5998–6008

Veličković P, Cucurull G, Casanova A et al (2018) Graph attention networks. In: International Conference on Learning Representations

Vijaikumar M, Hada D, Shevade S (2021) Hypertenet: Hypergraph and transformer-based neural network for personalized list continuation. In: 2021 IEEE International Conference on Data Mining (ICDM), IEEE, pp 1210–1215

Wang G, Ying R, Huang J et al (2019a) Improving graph attention networks with large margin-based constraints. arXiv preprint arXiv:1910.11945

Wang G, Ying R, Huang J et al (2021) Multi-hop attention graph neural networks. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)

Wang K, Shen W, Yang Y et al (2020a) Relational graph attention network for aspect-based sentiment analysis. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp 3229–3238

Wang P, Han J, Li C et al (2019b) Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 7152–7159

Wang X, He X, Cao Y et al (2019c) Kgat: Knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 950–958

Wang X, Ji H, Shi C et al (2019d) Heterogeneous graph attention network. In: The World Wide Web Conference, pp 2022–2032

Wang X, Zhu M, Bo D et al (2020b) Am-gcn: Adaptive multi-channel graph convolutional networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1243–1253

Wang X, Bo D, Shi C et al (2022) A survey on heterogeneous graph embedding: methods, techniques, applications and sources. IEEE Trans Big Data 9(2):415–436

Wang Y, Derr T (2021) Tree decomposed graph neural network. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp 2040–2049

Wang Z, Lei Y, Li W (2020) Neighborhood attention networks with adversarial learning for link prediction. IEEE Trans Neural Netw Learn Syst 32(8):3653–3663

Welling M, Kipf TN (2016) Semi-supervised classification with graph convolutional networks. In: J. International Conference on Learning Representations (ICLR 2017)

Wen Q, Zhou T, Zhang C et al (2022) Transformers in time series: A survey. arXiv preprint arXiv:2202.07125

Wu F, Souza A, Zhang T et al (2019a) Simplifying graph convolutional networks. In: International Conference on Machine Learning, PMLR, pp 6861–6871

Wu J, Shi W, Cao X et al (2021a) Disenkgat: knowledge graph embedding with disentangled graph attention network. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp 2140–2149

Wu L, Chen Y, Shen K et al (2021b) Graph neural networks for natural language processing: A survey. arXiv preprint arXiv:2106.06090

Wu Q, Zhang H, Gao X et al (2019b) Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In: The World Wide Web Conference, pp 2091–2102

Wu S, Sun F, Zhang W et al (2020) Graph neural networks in recommender systems: a survey. ACM Comput Surv (CSUR). https://doi.org/10.7717/peerj-cs.1166

Wu Z, Pan S, Chen F et al (2020) A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 32(1):4–24

Xia F, Sun K, Yu S et al (2021) Graph learning: a survey. IEEE Trans Artif Intell 2(2):109–127

Xia L, Huang C, Xu Y et al (2021b) Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 4486–4493

Xie Y, Zhang Y, Gong M et al (2020) Mgat: multi-view graph attention networks. Neural Netw 132:180–189

Xu D, Ruan C, Korpeoglu E et al (2020) Inductive representation learning on temporal graphs. arXiv preprint arXiv:2002.07962

Xu K, Hu W, Leskovec J et al (2018a) How powerful are graph neural networks? In: International Conference on Learning Representations

Xu K, Li C, Tian Y et al (2018b) Representation learning on graphs with jumping knowledge networks. In: International conference on machine learning, PMLR, pp 5453–5462

Xu X, Zu S, Gao C et al (2018c) Modeling attention flow on graphs. arXiv preprint arXiv:1811.00497

Xu Y, Wang L, Wang Y et al (2022) Adaptive trajectory prediction via transferable gnn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6520–6531

Yan S, Xiong Y, Lin D (2018) Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI Conference on Artificial Intelligence

Yang J, Liu Z, Xiao S et al (2021) Graphformers: Gnn-nested transformers for representation learning on textual graph. Adv Neural Inf Processing Syst 34:28,798-28,810

Yang L, Wu F, Gu J et al (2020) Graph attention topic modeling network. Proc Web Conf 2020:144–154

Yang L, Li M, Liu L et al (2021) Diverse message passing for attribute with heterophily. Adv Neural Inf Processing Syst 34:4751–4763

Yang M, Zhou M, Li Z et al (2022a) Hyperbolic graph neural networks: A review of methods and applications. arXiv preprint arXiv:2202.13852

Yang R, Shi J, Yang Y et al (2021) Effective and scalable clustering on massive attributed graphs. Proc Web Conf 2021:3675–3687

Yang S, Hu B, Zhang Z et al (2021) Inductive link prediction with interactive structure learning on attributed graph. Joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, pp 383–398

Yang T, Hu L, Shi C et al (2021) Hgat: heterogeneous graph attention networks for semi-supervised short text classification. ACM Trans Inf Syst(TOIS) 39(3):1–29

Yang Y, Wang X, Song M et al (2019) Spagan: shortest path graph attention network. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp 4099–4105

Yang Y, Qiu J, Song M et al (2020b) Distilling knowledge from graph convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7074–7083

Yang Y, Jiao L, Liu X et al (2022b) Transformers meet visual learning understanding: A comprehensive review. arXiv preprint arXiv:2203.12944

Yang Z, Dong S (2020) Hagerec: hierarchical attention graph convolutional network incorporating knowledge graph for explainable recommendation. Knowl-Based Syst 204(106):194

Ying C, Cai T, Luo S et al (2021) Do transformers really perform badly for graph representation? Adv Neural Inf Proc Syst 34:28,877-28,888

Ying Z, You J, Morris C et al (2018) Hierarchical graph representation learning with differentiable pooling. Adv Neural Inf Processing Systms. https://doi.org/10.48550/arXiv.1806.08804

Yuan H, Yu H, Gui S et al (2020) Explainability in graph neural networks: A taxonomic survey. arXiv preprint arXiv:2012.15445

Yuan H, Yu H, Wang J et al (2021a) On explainability of graph neural networks via subgraph explorations. In: International Conference on Machine Learning, PMLR, pp 12,241–12,252

Yuan J, Yu H, Cao M et al (2021b) Semi-supervised and self-supervised classification with multi-view graph neural networks. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp 2466–2476

Yun S, Jeong M, Kim R et al (2019) Graph transformer networks. Adv Neural Inf Processing Syst. https://doi.org/10.1016/j.neunet.2022.05.026

Zeng H, Zhou H, Srivastava A et al (2019) Graphsaint: Graph sampling based inductive learning method. In: International Conference on Learning Representations

Zhang C, Gao J (2021) Hype-han: Hyperbolic hierarchical attention network for semantic embedding. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pp 3990–3996

Zhang J, Shi X, Xie J et al (2018) Gaan: Gated attention networks for learning on large and spatiotemporal graphs. In: 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018

Zhang J, Zhang H, Xia C et al (2020a) Graph-bert: Only attention is needed for learning graph representations. arXiv preprint arXiv:2001.05140

Zhang J, Gao M, Yu J et al (2021a) Double-scale self-supervised hypergraph learning for group recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp 2557–2567

Zhang J, Chen Y, Xiao X et al (2022) Learnable hypergraph laplacian for hypergraph learning. ICASSP 2022–2022 IEEE International Conference on Acoustics. Speech and Signal Processing (ICASSP), IEEE, pp 4503–4507

Zhang J, Li F, Xiao X et al (2022b) Hypergraph convolutional networks via equivalency between hypergraphs and undirected graphs. arXiv preprint arXiv:2203.16939

Zhang K, Zhu Y, Wang J et al (2019) Adaptive structural fingerprints for graph attention networks. In: International Conference on Learning Representations

Zhang R, Zou Y, Ma J (2020b) Hyper-sagnn: a self-attention based graph neural network for hypergraphs. In: International Conference on Learning Representations (ICLR)

Zhang S, Xie L (2020) Improving attention mechanism in graph neural networks via cardinality preservation. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), NIH Public Access, p 1395

Zhang W, Chen Z, Dong C et al (2021b) Graph-based tri-attention network for answer ranking in cqa. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 14,463–14,471

Zhang W, Yin Z, Sheng Z et al (2022c) Graph attention multi-layer perceptron. arXiv preprint arXiv:2206.04355

Zhang X, Zeman M, Tsiligkaridis T et al (2021c) Graph-guided network for irregularly sampled multivariate time series. In: International Conference on Learning Representations (ICLR)

Zhang Y, Wang X, Shi C et al (2021) Hyperbolic graph attention network. IEEE Trans Big Data 8(6):1690–1701

Zhang Z, Zhuang F, Zhu H et al (2020c) Relational graph neural network with hierarchical attention for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 9612–9619

Zhao H, Wang Y, Duan J et al (2020) Multivariate time-series anomaly detection via graph attention network. In: 2020 IEEE International Conference on Data Mining (ICDM), IEEE, pp 841–850

Zhao Z, Gao B, Zheng VW et al (2017) Link prediction via ranking metric dual-level attention network learning. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp 3525–3531

Zheng C, Fan X, Wang C et al (2020) Gman: A graph multi-attention network for traffic prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 1234–1241

Zheng X, Liu Y, Pan S et al (2022) Graph neural networks for graphs with heterophily: A survey. arXiv preprint arXiv:2202.07082

Zhou J, Cui G, Hu S et al (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81

Zhou Y, Zheng H, Huang X et al (2022) Graph neural networks: taxonomy, advances, and trends. ACM Trans Intell Syst Technol (TIST) 13(1):1–54

Zhu Y, Lyu F, Hu C et al (2022) Learnable encoder-decoder architecture for dynamic graph: A survey. arXiv preprint arXiv:2203.10480

Zuo Y, Liu G, Lin H et al (2018) Embedding temporal network via neighborhood formation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 2857–2866

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.