Journal of Artificial Intelligence for Automation
https://www.sciltp.com/journals/jaia

Review

Graph Representation Learning in Complex Networks: Recent Advances and Open Challenges

Yue Yang [1,2,3], Dongxu Li [1,2,3], Ziwen Cui [1,2,3], Hengchuang Yin [1,2,3], Qinglong Sun [4], Mingjian Zhang [1,2,3], Ying Chang [1,2,3] and Lun Hu [1,2,3,*]

[1] Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi 830011, China
[2] University of Chinese Academy of Sciences, Beijing 101408, China
[3] Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China
[4] College of Computer Science and Technology, Xinjiang Normal University, Urumqi 830054, China
* Correspondence: hulun@ms.xjb.ac.cn

How To Cite: Yang, Y.; Li, D.; Cui,Z.; et al. Graph Representation Learning in Complex Networks: Recent Advances and Open Challenges. *Journal of Artificial Intelligence for Automation* **2026**, *1*(1), 4.

Received: 13 January 2026
Revised: 15 February 2026
Accepted: 26 February 2026
Published: 28 February 2026

Abstract: Complex networks have been widely adopted to model diverse real-world systems. However, the increasing complexity of such networks, characterized by heterogeneity, higher-order interactions, and temporal and dynamic learning settings, poses critical challenges to traditional machine learning methods. To address these challenges, graph representation learning (GRL) has emerged as a fundamental paradigm for analyzing complex networks by learning low-dimensional representations that preserve structural and semantic information in non-Euclidean spaces. With the rapid development of deep learning, graph neural networks (GNNs) have become central to GRL, enabling end-to-end learning on graph-structured data. In this survey, we provide a systematic and comprehensive overview of graph representation learning in complex networks. We review representation learning techniques at multiple granularities, including node-level, edge-level, and graph-level embeddings. In addition, we summarize key aspects of advanced GRL research, encompassing GNN architectural design, structural complexity modeling, learning and optimization strategies, as well as representative real-world applications. Finally, we highlight emerging topics and open challenges to outline promising directions for future research.

Keywords: complex networks; graph representation learning; graph neural networks

1. Introduction

Complex networks provide a unifying approach to systematically model a wide range of real-world systems, covering from social and information networks to biological interaction systems, transportation infrastructures, and cyber-physical systems [1–5]. From the network perspective, entities are naturally represented as nodes, and accordingly their relationships are regarded as edges. However, the existence of rich attributes, temporal dynamics, and multi-relational semantics further expands the analysis complexity of network-structured systems [6,7]. In these domains, the resulting network data are typically characterized as high-dimensional, sparse, heterogeneous, and non-Euclidean. Consequently, the comprehensive analysis of complex networks poses fundamental challenges to the applicability of traditional machine learning methods that rely on vectorized representations in Euclidean space [8,9].

In the last decades, Graph Representation Learning (GRL) has emerged as a promising deep learning framework to address these challenges by learning low-dimensional embeddings of nodes, edges, subgraphs, or entire graphs while preserving the underlying structural and semantic information [10–14]. Early GRL works focus on developing a variety of matrix factorization and random-walk-based methods by capturing network proximity [15–17]. Their contributions compose the significant foundation for efficient learning on large-scale complex networks. More recently, the rapid development of deep learning techniques has driven a research paradigm shift toward deep neural networks (DNNs) that directly operate on network-structured data. GNNs are representative DNN-based architectures in GRL, as they provide a principled message-passing framework that generalizes convolutional and attention mechanisms from Euclidean data to non-Euclidean graph domains [18–22].

Obviously, the increasing complexity of complex networks has further underscored the growing importance of GRL [23]. Real-world complex networks often exhibit heterogeneity across node and edge types [24], hierarchical and higher-order interactions [25], temporal structure patterns [26], and noisy or incomplete data [27]. These characteristics have motivated the development of advanced GRL techniques, including heterogeneous graph modeling [28], dynamic and temporal graph learning [29], hypergraph representations [30], as well as self-supervised and contrastive learning frameworks [31,32]. Meanwhile, critical concerns regarding scalability, interpretability, robustness, and theoretical understanding of GRL have become major topics in the research community. Hence, it is necessary to formulate a systematic and coherent survey that not only summarizes the state-of-the-art advances in GRL for complex network analysis, but also clarifies their strengths, limitations, and applications.

In this survey, rather than exhaustively presenting and analyzing GRL-based algorithms, we focus on summarizing the core ideas of different methodological families and on examining how these GRL-based methods evolve in response to increasing network complexity. Within this scope, we emphasize several key aspects that characterize existing research in GRL. First, we examine methodological paradigms, ranging from basic GNN architectures to more advanced models. Second, we consider structural complexity, including heterophily, long-range dependencies, higher-order relationships, and multi-relational or heterogeneous graph settings. Third, we address learning paradigms and optimization strategies, encompassing supervised, semi-supervised, and self-supervised learning, as well as causal learning approaches. Fourth, we review representative applications of graph representation learning techniques in real-world complex networks. Finally, we discuss emerging topics that are shaping the future of the field, including explainability, theoretical expressiveness, and the integration of domain knowledge. By achieving these objectives, we aim to guide readers from well-established foundations to current frontiers, while maintaining a coherent narrative that highlights the unifying principles of GRL techniques in complex networks.

The remainder of this survey is organized as follows. We begin by introducing preliminaries and problem formulations that establish common notation and define the key tasks in GRL. We then review classical GNN architectures. Subsequent sections focus on advanced graph representation learning methods, with particular emphasis on graph neural networks and their variants. A dedicated section is devoted to different learning paradigms and optimization strategies. Finally, we survey representative real-world applications across diverse domains and conclude with a discussion of open challenges and future research directions.

2. Preliminaries and Problem Formulation

In this section, the fundamental notations, network types, learning tasks, and evaluation protocols that underpin graph representation learning in complex networks are introduced. These preliminaries establish a unified formal framework for the diverse models and methodologies reviewed in subsequent sections. To ensure a clear and consistent formalization throughout the paper, we summarize the main notations and interpretations in Table 1.

Table 1. Main notations and interpretations used in this paper.

Notation	Interpretation
$\mathcal{G}$; $\mathcal{G}_s$; $\mathcal{G}_d$	A graph; Static graph; Dynamic graph
$\mathcal{G}_h$; $\mathcal{G}_{he}$; $\mathcal{G}_m$	Homogeneous, heterogeneous, and multiplex networks
$\mathcal{G}_a$; $\mathcal{G}_w$	Attributed graph; Weighted graph
V; n_{V}	Node set; Number of nodes
E; n_{E}	Edge set; Number of edges
Λ; n_{Λ}	Attribute set; Number of attributes
v_i; e_{ij}	The i-th node; Edge between v_i and v_j
$\mathrm{N}(v)$	Neighborhood of node v
$\mathrm{S}(i)$	Sampled neighbor set of node i
$\mathbf{A}$; $\mathbf{A}_w$; $\mathbf{X}$	Adjacency matrix; Weighted adjacency matrix; Feature matrix
$\tilde{\mathbf{A}}$; $\tilde{\mathbf{D}}$	Adjacency matrix with self-loops; Degree matrix of $\tilde{\mathbf{A}}$
$\mathbf{H}^{(l)}$	Node feature matrix at layer l
$\mathbf{W}^{(l)}$	Weight matrix at l-th layer (shared across nodes)
$\mathbf{m}_v^{(l)}$	Aggregated message vector of node v at layer l
$\mathbf{h}_v^{(l)}$	Node representation at l-th message passing layer
$\mathbf{h}_s$	Subgraph-level representation
$\mathbf{h}_{\mathcal{G}}$	Graph-level representation

Table 1. *Cont.*

Notation	Interpretation
$\mathcal{F}_{\text{node}}$	Node-level representation learning function
$\mathcal{F}_{\text{sub}}$	Subgraph-level representation learning function
$\mathcal{F}_{\text{graph}}$	Graph-level representation learning function
$s(v_i, v_j)$	Edge-level scoring function
$\text{AGGREGATE}^{(l)}(\cdot)$	Neighborhood aggregation function at layer l
$\text{UPDATE}^{(l)}(\cdot)$	Node update function at layer l
$\alpha_{ij}^{(l)}$	Attention coefficient from node j to node i at layer l
$\mathcal{L}_{\text{sup}}$; $\mathcal{L}_{\text{semi}}$	Supervised learning loss; Semi-supervised learning objective
$\mathcal{L}_{\text{pred}}$; $\mathcal{L}_{\text{gen}}$	Predictive self-supervised loss; Generative self-supervised loss
$\mathcal{L}_{\text{con}}$; $\mathcal{L}_{\text{gae}}$	Contrastive learning loss; Graph autoencoder reconstruction loss
$\mathcal{L}_{\text{mgae}}$	Masked graph autoencoder loss
$\mathcal{L}_{\text{crl}}$	Causal representation learning objective
ACC; Precision; Recall	Accuracy metric; Precision metric; Recall metric
F1; AUC	F1-score; Area Under the ROC Curve
NMI; ARI; Q	Normalized Mutual Information; Adjusted Rand Index; Modularity

2.1. Graph Notation and Network Types

Although real-world systems are naturally described as complex networks, most learning algorithms operate on graph-structured data defined by precise mathematical formalisms. Bridging this abstraction gap requires a clear distinction between graph notation, which specifies the mathematical representation, and network types, which characterize the structural and semantic properties of real-world systems.

Definition 1 (Graph). A graph is generally defined as $\mathcal{G} = (\text{V}, \text{E})$, where $\text{V} = \{v_i\}(1 \leq i \leq n_{\text{V}})$ is a set of n_{V} nodes, and $\text{E} = \{e_{ij}\}$ denotes the set of n_{E} edges. An edge $e_{ij} = (v_i, v_j) \in \text{E}$ indicates a relationship between nodes v_i and v_j. The neighborhood of a node v is defined as $\text{N}(v) = \{u \in \text{V} | (v, u) \in \text{E}\}$. Typically, we utilize an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n_{\text{V}} \times n_{\text{V}}}$ to characterize the graph structure.

2.1.1. Static vs. Dynamic Graphs

Based on whether the graph information evolves over time, graphs can be broadly categorized into static graphs and dynamic graphs.

Definition 2 (Static Graph). A static graph is a special case of Definition 1 in which the node set and edge set remain unchanged throughout the modeling process. Formally, a static graph is denoted as $\mathcal{G}_s = (\text{V}, \text{E})$, where both V and E are fixed. Accordingly, the associated adjacency matrix $\mathbf{A} \in \mathbb{R}^{n_{\text{V}} \times n_{\text{V}}}$ is constant.

Definition 3 (Dynamic Graph). By introducing an explicit time dimension, a dynamic graph explicitly models the temporal evolution of graph-structured data. It is defined as $\mathcal{G}_d = (\text{V}(t), \text{E}(t))$, where $\text{V}(t)$ and/or $\text{E}(t)$ may vary over time. In discrete-time models, a dynamic graph can be represented as a sequence of snapshots $\{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(T)}\}$, while in continuous-time models, temporal information is captured via timestamps associated with edges: $\text{E} = \{(v_i, v_j, t_k) | v_i, v_j \in \text{V}, t_k \in \mathbb{R}^+\}$.

2.1.2. Homogeneous, Heterogeneous, and Multiplex Networks

From the perspective of semantic granularity and relational diversity, networks can be categorized according to whether nodes and edges are associated with multiple types and how such relational heterogeneity is organized. Based on this viewpoint, networks are commonly categorized into homogeneous networks, heterogeneous networks, and multiplex networks.

Definition 4 (Homogeneous Network). A homogeneous network refers to a graph in which all nodes and edges are assumed to share a unified semantic interpretation, implying a single node type and a single relation type throughout the network. Formally, it is denoted as $\mathcal{G}_h = (\text{V}, \text{E})$, where each node $v_i \in \text{V}$ and each edge $e_{ij} \in \text{E}$ belongs to the same semantic type.

Definition 5 (Heterogeneous Network). A heterogeneous network generalizes Definition 1 by allowing nodes and/or edges to belong to multiple semantic types, thereby encoding richer relational and entity information. Formally, it is represented as $\mathcal{G}_{he} = (\mathrm{V}, \mathrm{E}, \phi, \psi)$, where $\phi : \mathrm{V} \to \mathcal{T}_{\mathrm{V}}$ assigns a type to each node, and $\psi : \mathrm{E} \to \mathcal{T}_{\mathrm{E}}$ assigns a type to each edge. Here, $\mathcal{T}_{\mathrm{V}}$ and $\mathcal{T}_{\mathrm{E}}$ denote the sets of all possible node types and edge types, respectively.

Definition 6 (Multiplex Network). A multiplex network is a network in which multiple types of relations among the same set of nodes are organized into separate layers. Formally, a multiplex network is denoted as $\mathcal{G}_m = (\mathrm{V}, \{\mathrm{E}^{(l)}\}_{l=1}^{L})$, where all layers share the same node set V, and each edge set $\mathrm{E}^{(l)} \subseteq \mathrm{V} \times \mathrm{V}$ represents a distinct type of relation.

2.1.3. Attributed and Weighted Graphs

Graphs can be enriched by associating additional information with nodes or edges, which gives rise to attributed and weighted graphs.

Definition 7 (Attributed Graph). An attributed graph builds on the basic graph of Definition 1 by associating nodes with attribute information. Formally, it is represented as $\mathcal{G}_a = (\mathrm{V}, \mathrm{E}, \Lambda)$, where $\Lambda = \{\Lambda_m\}(1 \leq m \leq n_\Lambda)$ denotes the set of n_Λ attributes associated nodes in V. Typically, we utilize a feature matrix $\mathbf{X} \in \mathbb{R}^{n_{\mathrm{V}} \times n_\Lambda}$ to characterize the attribute information.

Definition 8 (Weighted Graph). A weighted graph is a graph as in Definition 1, in which each edge is assigned a scalar weight to represent the strength or importance of the relationship. Formally, $\mathcal{G}_w = (\mathrm{V}, \mathrm{E}, \mathbf{A}_w)$, where $\mathbf{A}_w \in \mathbb{R}^{n_{\mathrm{V}} \times n_{\mathrm{V}}}$ is the weighted adjacency matrix with $a_{w_{ij}} > 0$ representing the weight of edge e_{ij}.

2.2. Representation Learning Tasks

Graph representation learning aims to encode graph-structured data into low-dimensional embeddings that preserve task-relevant structural and semantic information. Depending on the target object and supervision granularity, existing methods can be broadly categorized into the following four classes.

- **Node-level learning** focuses on learning an embedding vector for each node in the graph. Formally, it aims to learn a mapping
$$\mathcal{F}_{\mathrm{node}} : v_i \in \mathrm{V} \to \mathbf{h}_i \in \mathbb{R}^d, \tag{1}$$
where $\mathbf{h}_i$ encodes both the local neighborhood structure and node attributes, and d denotes the dimensionality of the node embedding. This task underlies a wide range of applications, including node classification, node clustering, and anomaly detection. The key challenge lies in preserving structural proximity and semantic consistency among nodes under complex network dependencies.
- **Edge-level learning** aims to model pairwise relationships between nodes, typically through a scoring or representation function
$$s(v_i, v_j) = \mathcal{G}(\mathbf{h}_i, \mathbf{h}_j), \tag{2}$$
where $\mathcal{G}(\cdot)$ denotes an edge-level scoring function, and $s(v_i, v_j)$ represents the resulting score for node pair (v_i, v_j), which estimates the existence, strength, or type of an edge between v_i and v_j. Representative applications include link prediction, relation classification, and interaction inference, where relational patterns rather than individual entities are the primary learning targets.
- **Subgraph-level learning** targets mesoscopic structures by encoding an induced subgraph $\mathcal{G}_s = (\mathrm{V}_s, \mathrm{E}_s)$ into a fixed-dimensional vector
$$\mathcal{F}_{\mathrm{sub}} : \mathcal{G}_s \to \mathbf{h}_s \in \mathbb{R}^d. \tag{3}$$
This formulation is particularly suitable for capturing higher-order structural patterns such as motifs, communities, and functional modules, which are not adequately represented at the node or edge level.
- **Graph-level learning** aims to learn a holistic representation for an entire graph
$$\mathcal{F}_{\mathrm{graph}} : \mathcal{G} \to \mathbf{h}_{\mathcal{G}} \in \mathbb{R}^d, \tag{4}$$
where $\mathbf{h}_{\mathcal{G}}$ summarizes global structural and semantic properties. This task is central to applications such as graph classification, regression, and similarity retrieval, especially in scenarios where each graph instance is treated as an independent sample.

These tasks form a hierarchical spectrum in terms of representation granularity, ranging from local entity-level modeling to global graph-level abstraction. In practice, many graph learning models jointly optimize multiple levels to enhance representation expressivity and task performance.

2.3. Evaluation Protocols

Evaluation protocols play a critical role in assessing the effectiveness and robustness of graph representation learning methods. In this section, we summarize the experimental evaluation in terms of commonly used benchmark datasets and the corresponding experimental settings and performance metrics.

2.3.1. Common Benchmark Datasets

Benchmark datasets used in graph representation learning span multiple domains and network types, reflecting the heterogeneity of real-world complex systems. In this work, we mainly categorize them into six groups, i.e., synthetic networks, citation networks, social networks, biochemical networks, product co-purchasing networks, and air-traffic networks. The detailed descriptions of each category are provided below.

- **Synthetic networks** are widely adopted to systematically evaluate algorithmic behavior under controlled conditions. The Girvan-Newman (GN) [33] generates networks with predefined community structures by controlling the ratio of intra- and inter-community connections. The Lancichinetti-Fortunato-Radicchi (LFR) [34] provides a more realistic simulation by modeling power-law degree and community size distributions, and supports extensions to weighted, directed, and overlapping community settings.
- **Citation networks** model scholarly communication and collaboration patterns. In such networks, nodes typically represent publications or authors, and edges denote citation or co-authorship relationships. Representative datasets include Cora [35], Citeseer [35], and Pubmed [35], which are widely used as attributed graphs where nodes are described by bag-of-words or binary word vectors and labels correspond to research topics. More complex scenarios are captured by datasets such as DBLP [36] and ACM [37], which can be constructed as heterogeneous or multi-view networks by incorporating multiple entity types (e.g., papers, authors, venues) and relations (e.g., authorship, citation, topical similarity).
- **Social networks** describe interactions among users in social media and online platforms, where nodes represent users and edges encode relationships such as friendships or followings. Popular benchmarks include YouTube [36], LiveJournal [38], and PolBlogs [39], which provide ground-truth communities defined by user groups, interests, or political affiliations. Attributed social networks further enrich this setting. For example, Facebook [40], Twitter [40], Gplus [40], Flickr [41], and BlogCatalog [41] incorporate user profiles, content tags, or interaction metadata as node attributes.
- **Biochemical networks** represent molecules, chemical compounds, or proteins as graphs, where nodes correspond to atoms or proteins and edges denote chemical bonds or protein–protein interactions. These networks are commonly employed to predict molecular properties, chemical activity, or protein functions, reflecting their relevance in bioinformatics and cheminformatics applications. Representative benchmark datasets include NCI-1 [42], MUTAG [43], PROTEINS [44], PTC [45], QM9 [46], Alchemy [47], and the Protein-Protein Interaction (PPI) network [48].
- **Product co-purchasing networks** capture patterns of user consumption, where nodes represent products and edges indicate frequent co-purchases. These networks are widely used to study product similarity, community structures, and to support recommendation systems. Representative benchmark datasets include Amazon [36] and Polbooks [49], in which ground-truth communities correspond to product categories or user preference groups.
- **Air-traffic networks** model the connectivity of airports through direct flights, with nodes representing airports and edges representing flight routes. These networks are often used to study network efficiency, flow optimization, and community structure. Representative datasets include Brazil Air-Traffic (BAT) [50], Europe Air-Traffic (EAT) [50], and USA Air-Traffic (UAT) [50].

2.3.2. Experimental Settings and Performance Metrics

In practice, graph representation learning models are typically implemented using popular deep learning frameworks and libraries, such as PyTorch Geometric and Deep Graph Library (DGL), which provide efficient implementations of various GRL architectures. In addition to implementation, performance metrics play a crucial role in assessing model effectiveness. In the following, we summarize commonly adopted performance metrics for evaluating graph representation learning methods. These metrics are categorized by task type and are described as follows.

- **Accuracy (ACC).** Accuracy quantifies the overall correctness of predictions by measuring the proportion of instances whose predicted labels match the ground truth. Given predicted labels $\{y_i\}_{i=1}^{n_\mathrm{V}}$ and ground-truth labels $\{y_i^*\}_{i=1}^{n_\mathrm{V}}$, ACC is computed as

$$\mathrm{ACC} = \frac{1}{n_\mathrm{V}} \sum_{i=1}^{n_\mathrm{V}} \mathbb{I}(y_i = y_i^*), \tag{5}$$

where $\mathbb{I}(\cdot)$ is the indicator function. This metric is widely used for node-level and graph-level classification tasks where overall prediction accuracy is of interest.

- **Precision.** Precision measures the reliability of positive predictions by evaluating the fraction of correctly predicted positive instances among all instances predicted as positive:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{6}$$

where TP and FP denote the numbers of true positives and false positives, respectively. Precision is particularly informative in scenarios with class imbalance or when false positives carry higher costs.

- **Recall.** Recall assesses the ability to identify all relevant positive instances, defined as

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{7}$$

where FN is the number of false negatives. This metric is commonly used in imbalanced classification tasks or applications where missing positive instances is critical.

- **F1-score (F1).** The F1-score provides a single measure that balances Precision and Recall, capturing both the correctness and completeness of positive predictions:

$$\mathrm{F1} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{8}$$

It is widely adopted in node-level classification and edge prediction tasks to provide a robust evaluation when class distributions are uneven.

- **Area Under the ROC Curve (AUC).** The AUC evaluates the ranking quality of predicted scores by estimating the probability that a randomly selected positive instance is assigned a higher score than a randomly selected negative instance:

$$\mathrm{AUC} = \mathbb{P}(s^+ > s^-), \tag{9}$$

where s^+ and s^- denote the predicted scores for positive and negative instances, respectively. This metric is widely used in binary classification tasks and is particularly suitable for edge-level applications such as link prediction.

- **Normalized Mutual Information (NMI).** The main principle of NMI is to measure the degree of similarity between the identified clusters and ground truth by comparing all possible combinations of them in a quantitative manner. Assuming that $\mathrm{Z} = \mathrm{Z}_i(1 \le i \le k)$ and $\mathrm{C} = \mathrm{C}_i(1 \le i \le k)$ denote the ground-truth and predicted clusters, respectively, where k is the number of clusters. Then, the NMI metric can be defined as:

$$\mathrm{NMI} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} n_{\mathrm{C}_i,\mathrm{Z}_j} log(\frac{n_\mathrm{V} n_{\mathrm{C}_i,\mathrm{Z}_j}}{n_{\mathrm{C}_i} n_{\mathrm{Z}_j}})}{\sqrt{(\sum_{i=1}^{k} n_{\mathrm{C}_i} log\frac{n_{\mathrm{C}_i}}{n_\mathrm{V}})(\sum_{j=1}^{k} n_{\mathrm{Z}_j} log\frac{n_{\mathrm{Z}_j}}{n_\mathrm{V}})}} \tag{10}$$

where n_{C_i} is the number of nodes in n_{C_i}, n_{Z_i} is the number of nodes in n_{Z_i}, and $n_{\mathrm{C}_i,\mathrm{Z}_j}$ is the number of nodes found in both n_{C_i} and n_{C_i}. NMI is widely used for evaluating unsupervised or semi-supervised node clustering quality.

- **Adjusted Rand Index (ARI).** ARI measures the agreement between the predicted clustering and the ground truth by considering all pairs of nodes, while explicitly correcting for agreement that may occur by random chance. Assuming that $\mathrm{Z} = \{\mathrm{Z}_i\}_{i=1}^{k}$ and $\mathrm{C} = \{\mathrm{C}_i\}_{i=1}^{k}$ denote the ground-truth and predicted clusters, respectively, where k is the number of clusters. Let $n_{\mathrm{C}_i,\mathrm{Z}_j}$ denote the number of nodes that simultaneously belong to cluster C_i and Z_j, and n_V be the total number of nodes. The ARI is defined as

$$\mathrm{ARI}=\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\binom{n_{\mathrm{C}_i,\mathrm{Z}_j}}{2}-\frac{\sum_{i=1}^{k}\binom{n_{\mathrm{C}_i}}{2}\sum_{j=1}^{k}\binom{n_{\mathrm{Z}_j}}{2}}{\binom{n_{\mathrm{V}}}{2}}}{\frac{1}{2}\left[\sum_{i=1}^{k}\binom{n_{\mathrm{C}_i}}{2}+\sum_{j=1}^{k}\binom{n_{\mathrm{Z}_j}}{2}\right]-\frac{\sum_{i=1}^{k}\binom{n_{\mathrm{C}_i}}{2}\sum_{j=1}^{k}\binom{n_{\mathrm{Z}_j}}{2}}{\binom{n_{\mathrm{V}}}{2}}}. \tag{11}$$

ARI is commonly used to evaluate node clustering performance induced by learned representations, and provides a more reliable assessment than the Rand Index by discounting chance-level agreement.

- **Modularity (Q).** Modularity quantifies the structural quality of a graph partition by measuring the extent to which intra-community edges are denser than expected under a random null model. For an undirected and unweighted graph, it is defined as

$$\mathrm{Q}=\sum_{k}(e_{kk}-a_k^2), \tag{12}$$

where e_{kk} denotes the fraction of edges within the k-th group and a_k represents the fraction of edges incident to nodes in that group. Modularity is primarily used in unsupervised community detection and graph clustering tasks to evaluate the structural coherence of the resulting partitions.

3. Graph Neural Networks: Foundations and Architectures

Graph neural networks (GNNs) were first introduced in the mid-2000s [51] and have since emerged as a central paradigm for learning on graph-structured data [52]. Graph structures provide a highly expressive and powerful representation, and under the modern deep learning paradigm, a wide range of problems can be naturally formulated as graph-structured data. In Figure 1, we illustrate two special cases of graph-structured representations.

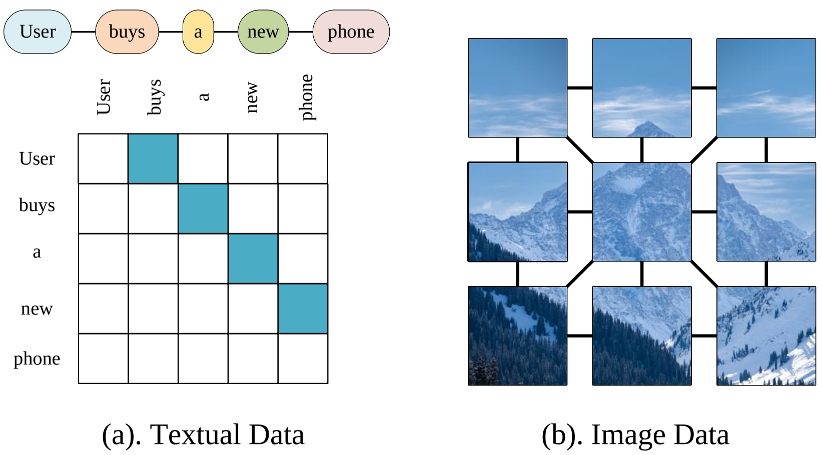

Figure 1. (**a**) illustrates the transformation of textual data into a graph-structured representation, while (**b**) shows how image data are converted into graph structures based on pixel-level information.

For textual data, a fully connected "word graph" can be constructed, where words are treated as nodes and the sequential order between words is represented as directed edges, resulting in a simple directed graph structure. For image data, images are typically viewed as matrix-like grids with multiple channels. Within the graph neural network framework, an image can be interpreted as a graph with a fixed and regular topology, where each pixel corresponds to a node and edges are defined based on spatial adjacency. Each non-boundary pixel has exactly eight neighboring pixels, and the information stored at each node can be represented as a three-dimensional vector corresponding to the RGB color values.

The above two examples belong to Euclidean-structured [53] data with highly regular patterns. As a result, existing methods typically do not employ GNNs to process these data types. For textual sequence data, recurrent neural networks (RNNs) [54] are commonly used to capture global dependencies and to model the probability of the next token. For image data, convolutional neural networks (CNNs) [55] apply fixed-size $n \times n$ filters, where neighborhood relationships are predefined and invariant. By sliding convolutional filters over the image grid, local feature extraction is efficiently performed.

However, traditional RNNs and CNNs are not well suited for handling non-Euclidean graph-structured data, such as social networks [56], molecular structures [57], and knowledge graphs [58], as illustrated in Figure 2. These graphs are characterized by irregular topologies and variable-sized neighborhoods. To address such challenges, GNNs are capable of aggregating information from neighboring nodes in complex graph structures and demonstrate strong learning capabilities for non-Euclidean data. In this section, we systematically introduce how GNNs perform information aggregation through the General Message Passing Framework [59], and on this basis, we review representative GNN architectures proposed in recent years.

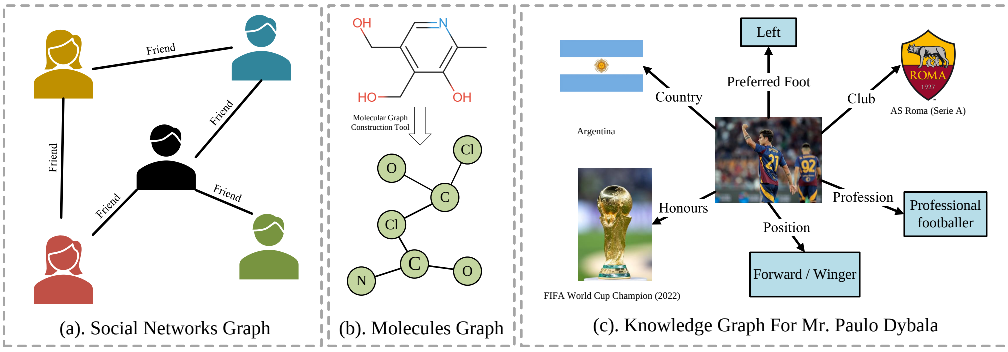

Figure 2. Examples of graph-structured data: (**a**) a social network graph with users as nodes and directed or undirected edges depending on interaction types; (**b**) a molecular graph constructed from drug molecules, where nodes represent atoms and edges denote chemical bonds; (**c**) a knowledge graph, where nodes represent attributes and relations encode semantic connections.

3.1. General Message Passing Framework

To enable effective learning on graphs, the message passing paradigm is employed to model complex graph-structured data. Message passing allows for learnable transformations over all components of a graph, including nodes, edges, and global context, while preserving permutation invariance. As a result, graph neural networks (GNNs) do not alter the connectivity of the input graph; instead, they retain the original adjacency structure and update node, edge, or graph-level feature representations.

Specifically, information from neighboring nodes and edges is aggregated and encoded into a single node representation, which is then used to update the node's own features. In subsequent layers, the updated representations further propagate information based on their respective neighborhoods, and this process is repeated iteratively. When a sufficient number of layers is stacked, GNNs are theoretically capable of capturing information from all nodes in large-scale graph-structured data. As illustrated in Figure 3, a simple three-layer graph neural network is designed to visualize how node representations in the final layer progressively accumulate information from surrounding nodes across layers. In this design, three layers are sufficient to enable each node to incorporate information from the entire graph.

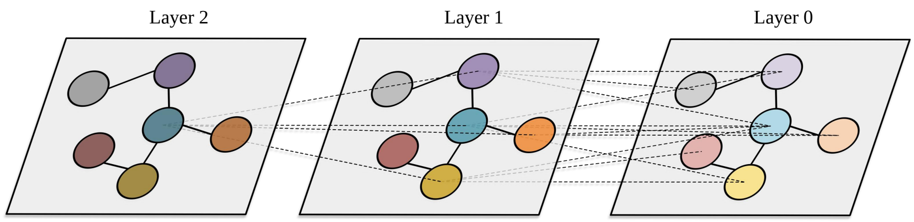

Figure 3. A graph neural network with two message-passing layers (Layer 1 and Layer 2), where node representations are iteratively updated from the input features (Layer 0).

The message passing process in GNNs can be formalized as neighborhood aggregation and update rules. Under this learning paradigm, GNNs have demonstrated strong predictive performance across a wide range of application domains, including drug–target interaction prediction [60], social network analysis [61], recommender systems [62], and knowledge graph reasoning.

Neighborhood Aggregation and Update Rules

In each message passing iteration, every node receives and aggregates information from its neighboring nodes in order to update its own feature representation. Formally, let $\mathbf{h}_v^{(l)}$ denote the representation of node v at the l-th iteration, where v represents a node in the graph and l denotes the message passing layer or iteration step. When $l = 1$, each node aggregates information only from its one-hop neighbors; as the number of iterations increases (e.g., $l = 2$), the node representation further incorporates information from its two-hop neighborhood. With increasing network depth, node features progressively encode structural and attribute information from more distant nodes, enabling the modeling of long-range dependencies across the entire graph.

Within the message passing framework of graph neural networks (GNNs), neighborhood aggregation and update rules constitute the core mechanism for node representation learning. This process typically consists of two stages: message aggregation and node update. Specifically, for a node v, the aggregated message received at layer l is defined as

$$\mathbf{m}_v^{(l)} = \mathrm{AGGREGATE}^{(l)} \left(\left\{ \mathbf{h}_u^{(l-1)} \mid u \in \mathcal{N}(v) \right\} \right), \tag{13}$$

where $\mathbf{m}_v^{(l)}$ denotes the aggregated message vector of node v at the l-th layer, $\mathcal{N}(v)$ represents the set of neighboring nodes of v, u denotes an arbitrary neighbor of v, and $\mathbf{h}_u^{(l-1)}$ is the feature representation of neighbor node u from the previous layer. The function $\mathrm{AGGREGATE}^{(l)}(\cdot)$ denotes the neighborhood aggregation function used at layer l. Since its input is an unordered set of node features, the aggregation function must satisfy permutation invariance. Common aggregation operations include summation, mean pooling, and max pooling.

After obtaining the aggregated message, the node representation is iteratively updated through an update function defined as

$$\mathbf{h}_v^{(l)} = \mathrm{UPDATE}^{(l)} \left(\mathbf{h}_v^{(l-1)}, \mathbf{m}_v^{(l)} \right), \tag{14}$$

where $\mathbf{h}_v^{(l)}$ denotes the updated representation of node v at layer l, $\mathbf{h}_v^{(l-1)}$ is its representation from the previous layer, and $\mathrm{UPDATE}^{(l)}(\cdot)$ represents the node update function at layer l. This function is typically implemented using learnable neural network modules, such as linear transformations followed by nonlinear activation functions, or by incorporating gating mechanisms to control the fusion of new and existing information.

Through the multi-layer stacking of the above neighborhood aggregation and update processes, node representations are able to progressively integrate structural and feature information from increasingly larger graph neighborhoods. As a result, graph neural networks can effectively model complex node dependencies in non-Euclidean graph data. This general message passing paradigm provides a unified theoretical foundation for subsequent spatial-domain graph neural network models.

3.2. Spatial-Based Neural Models

Spatial-based [63] neural models constitute one of the most prevalent modeling paradigms in graph neural networks. The key idea of this paradigm is to perform learning directly on the original graph topology. Since graphs do not possess a fixed node ordering, the model should not rely on any predefined node order during training or inference. By stacking multiple layers, GNNs are able to capture long-range dependencies inherent in spatial graph data.

A substantial body of existing research focuses on leveraging GNNs to model complex spatial or spatio-temporal dependencies in spatial data. According to different neighborhood aggregation strategies, spatial-based models can be broadly categorized into three major types.

Convolutional-Based, Attention-based, and Sampling-based GNNs

Convolutional-Based: Graph neural networks are inspired by convolutional neural networks. As illustrated in Figure 4, for regular grid-like structures, a fixed-size convolutional kernel can be applied by introducing padding around each node, resulting in an output graph with the same size as the input, or alternatively by reducing the dimensionality through convolution. Convolutional layers operate on small local receptive fields and learn hierarchical representations. However, fixed convolution kernels cannot be directly applied to non-Euclidean graph-structured data. To address this limitation, graph convolution techniques were introduced to specifically handle graph-structured data, leading to the development of graph neural networks.

Graph Convolutional Networks (GCNs) [64] are one of the fundamental variants of graph neural networks. The core idea of GCNs is to perform convolution-like operations on graph data by jointly considering the features of a node and those of its neighboring nodes, thereby enabling effective information propagation across the graph. GCNs are typically composed of multiple layers, where each layer updates node representations through convolution and aggregation operations. To date, GCNs have been successfully applied to a wide range of real-world problems, including image segmentation [65] and recommender systems [62].

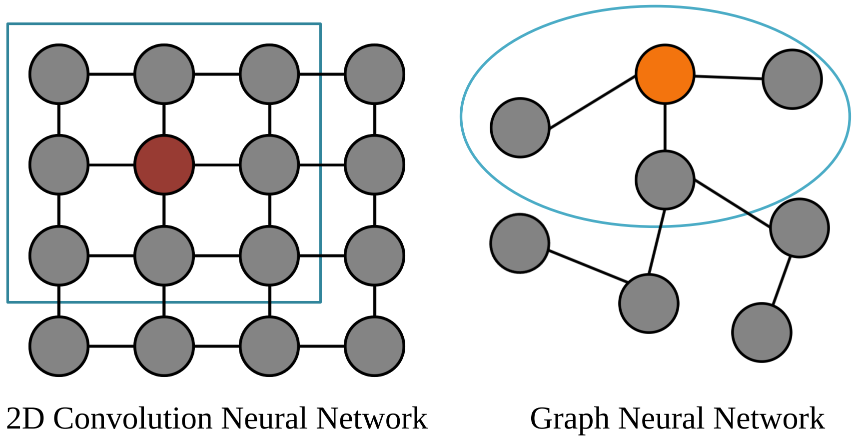

Figure 4. The **left panel** illustrates the application of convolution on regular structured data (e.g., image pixels), while the **right panel** shows convolution applied to non-Euclidean data structures.

A commonly used formulation updates the representation of node u at layer $l+1$ as

$$\mathbf{h}_u^{(l+1)} = \sigma\left(\sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{1}{\sqrt{\tilde{d}_u \tilde{d}_v}} \mathbf{W}^{(l)} \mathbf{h}_v^{(l)}\right) \tag{15}$$

where $\mathbf{h}_u^{(l)}$ denotes the feature representation of node u at the l-th layer, $\mathcal{N}(u)$ is the set of one-hop neighbors of u, and $\tilde{d}_u$ denotes the degree of node u after adding self-loops. The weight matrix $\mathbf{W}^{(l)}$ is shared across all nodes at layer l, and $\sigma(\cdot)$ is a nonlinear activation function.

By stacking multiple such layers, node representations progressively incorporate information from larger receptive fields. The node-wise update in Equation (15) can be equivalently expressed in a compact matrix form as

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)}\right) \tag{16}$$

where $\mathbf{H}^{(l)} \in \mathbb{R}^{|\mathcal{V}| \times d_l}$ is the node feature matrix at layer l, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, and $\tilde{\mathbf{D}}$ is the corresponding degree matrix with $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$.

Equations (15) and (16) illustrate that convolutional-based GNNs can be viewed as a specific instantiation of the message passing framework, where neighborhood aggregation is realized through degree-normalized feature propagation.

Attention-Based: Graph Attention Networks (GATs) [66,67] are neural network architectures specifically designed for processing and analyzing graph-structured data. As illustrated in Figure 5, GAT can be regarded as a variant of Graph Convolutional Networks (GCNs), in which attention mechanisms—originally introduced in sequence modeling—are incorporated into the neighborhood aggregation process. Instead of treating all neighboring nodes equally, GAT dynamically assigns different importance weights to each neighbor. By computing attention scores for neighboring nodes, GAT is able to identify which nodes are more relevant during aggregation, thereby enabling more expressive and adaptive feature learning.

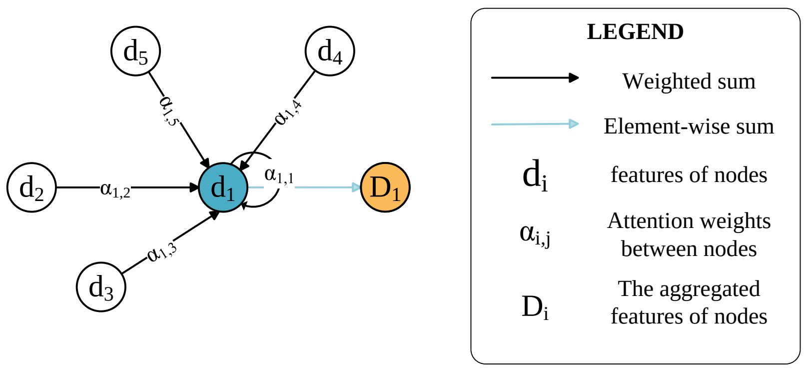

Figure 5. Illustration of Graph Attention Networks (GAT), where neighboring nodes and edges are assigned different attention scores based on their relative importance.

GAT models have now been widely applied in the field of bioinformatics [68], particularly for analyzing heterogeneous graph data in biological networks. Representative application domains include drug discovery [69], protein–protein interaction analysis [70], and cancer research [71].

In graph attention networks, node representations are updated by assigning learnable attention weights to neighboring nodes. For node i, the attention-based aggregation at layer $l+1$ is defined as

$$\alpha_{ij}^{(l)} = \frac{\exp\Big(a\Big(\mathbf{W}^{(l)}\mathbf{h}_i^{(l)}, \mathbf{W}^{(l)}\mathbf{h}_j^{(l)}\Big)\Big)}{\sum_{k\in\mathcal{N}(i)} \exp\Big(a\Big(\mathbf{W}^{(l)}\mathbf{h}_i^{(l)}, \mathbf{W}^{(l)}\mathbf{h}_k^{(l)}\Big)\Big)} \tag{17}$$

where i denotes the target node and j (or k) denotes a first-order neighbor of node i, i.e., $j, k \in \mathcal{N}(i)$, with $\mathrm{N}(i)$ being the 1-hop neighborhood set of node i. The superscript l indicates the layer index. $\mathbf{h}_i^{(l)} \in \mathbb{R}^{d_l}$ is the feature representation (embedding) of node i at layer l, and $\mathbf{W}^{(l)} \in \mathbb{R}^{d_{l+1}\times d_l}$ is a shared trainable linear transformation matrix. The attention function $a(\cdot,\cdot)$ computes an unnormalized importance score between the transformed features of nodes i and j, and $\alpha_{ij}^{(l)}$ is the corresponding normalized attention coefficient obtained by the softmax operation, satisfying $\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(l)} = 1$. Here, $\exp(\cdot)$ is the exponential function used in softmax normalization.

The node representation is then updated by a weighted aggregation of neighbor features:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(l)}\mathbf{W}^{(l)}\mathbf{h}_j^{(l)}\right) \tag{18}$$

where $\sigma(\cdot)$ denotes a nonlinear activation function and $\mathbf{h}_i^{(l+1)} \in \mathbb{R}^{d_{l+1}}$ is the updated representation of node i at layer $l+1$, obtained by aggregating the linearly transformed neighbor features $\mathbf{W}^{(l)}\mathbf{h}_j^{(l)}$ weighted by the attention coefficients $\alpha_{ij}^{(l)}$.

Sampling-Based: Under the sampling-based paradigm [72,73], a representative model is GraphSAGE [74] . As illustrated in Figure 6, GraphSAGE samples a fixed-size set of neighbors for each node rather than aggregating all adjacent nodes, thereby avoiding excessive computation and memory costs on large-scale graphs and enabling scalable graph representation learning. In contrast, GCN and GAT typically aggregate over the full neighborhood at each layer. However, not every neighbor contributes equally to the central node, and blindly performing full aggregation and stacking multiple layers on large graphs can incur substantial computational and storage overhead. GraphSAGE approximates the overall neighborhood distribution by sampling a fixed number of neighbors per layer, achieving a practical trade-off between efficiency and cost for very large graphs.

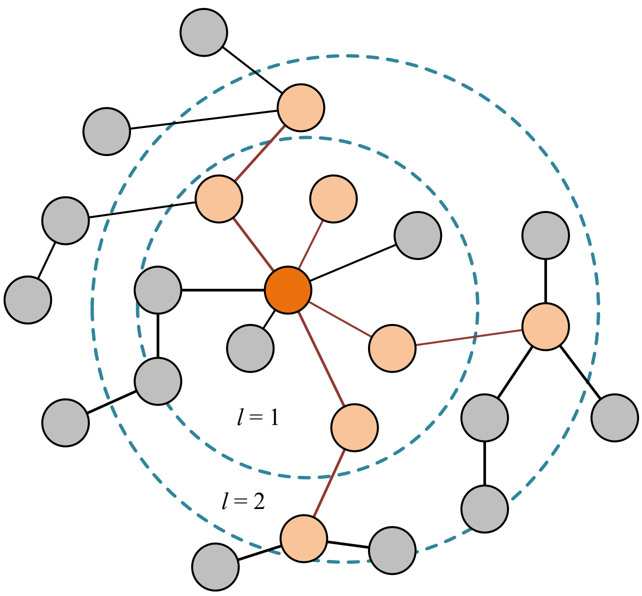

Figure 6. In large-scale graphs, a subset of neighbor nodes is sampled independently at each layer. The central dark-orange node is the target, light-orange nodes indicate sampled neighbors, gray nodes are un-sampled, and red edges indicate edges used for aggregation. This layer-wise sampling reduces computation while preserving multi-hop information.

A standard GraphSAGE update can be written as follows:

$$\mathbf{h}_i^{(l+1)} = \sigma\Big(\mathbf{W}^{(l)} \cdot \mathrm{AGGREGATE}\Big(\Big\{\mathbf{h}_j^{(l)} \mid j \in \mathcal{S}(i)\Big\}\Big)\Big), \tag{19}$$

where $\mathbf{h}_i^{(l)}$ denotes the feature representation of node i at layer l, $\mathrm{S}(i)$ denotes the sampled neighbor set of node i, $\mathbf{W}^{(l)}$ is a trainable weight matrix shared across nodes at layer l, and $\sigma(\cdot)$ is a nonlinear activation function. The aggregation function $\mathrm{AGGREGATE}(\cdot)$ is permutation-invariant and can be instantiated as mean-based, pooling-based, or LSTM-based aggregation to capture complex neighborhood dependencies.

3.3. Limitations

Although graph neural networks (GNNs) have shown strong performance in modeling non-Euclidean data, their practical deployment still faces several challenges [75,76]. One challenge lies in the complexity of graph construction and the preprocessing required before training. Moreover, since GNNs fundamentally rely on aggregating information from neighboring nodes, the presence of noise in the data can impose additional limitations on model learning and generalization. In this section, we summarize three major limitations of GNNs: oversmoothing, oversquashing, and depth constraints.

Oversmoothing, Oversquashing, and Depth Constraints

Oversmoothing: In traditional GNNs, after multiple message passing layers, node representations may become overly similar. For example, consider two nodes A and B whose graph distance is within k hops. Since each message passing layer performs similar aggregation operations, neighborhood information is repeatedly accumulated. When the number of layers exceeds k, the feature representations of nodes A and B tend to become increasingly similar as depth grows. This suggests that oversmoothing is more likely to occur in smaller graphs with stronger connectivity.

Oversquashing: Message passing is intended to enable a node to incorporate information from distant nodes. However, as the number of nodes associated with a target node v increases, a GNN must compress a large amount of node/edge information into a fixed-dimensional vector representation for v. Representing massive information with limited vector dimensionality inevitably leads to information loss. In particular, information from distant nodes can be severely compressed when it reaches the central node, which may degrade downstream performance. This also indicates that modeling long-range dependencies remains challenging and is a key limitation of GNNs.

Depth Constraints: Due to the above two issues, a shallow GNN may be insufficient to aggregate information from distant or peripheral regions of the graph, whereas an overly deep GNN can cause node representations to become indistinguishable. As a result, in practical applications, increasing depth beyond a certain point may instead lead to performance degradation.

It is worth noting that these limitations are interrelated: oversquashing can exacerbate oversmoothing by compressing distant information, while depth constraints reflect the practical trade-off between mitigating oversquashing and avoiding oversmoothing. In practice, these factors may compound each other, collectively impacting GNN performance.

4. Graph Representation Learning For Complex Network Structures

Graph Representation Learning (GRL) aims to map graph-structured data into low-dimensional continuous embeddings while preserving the structural, semantic, and temporal signals needed by downstream tasks. Early GRL approaches predominantly focus on static and homogeneous graphs, operating under simplifying assumptions such as a single node type, a single relation type, and a time-invariant topology [77–79]. In contrast, real-world networks—including knowledge graphs, biological interaction networks, and social systems—exhibit substantially richer characteristics [80,81], where semantic heterogeneity, higher-order organization, and temporal evolution are often intrinsic rather than incidental.

From a representation-learning standpoint, these complexities induce three corresponding modeling requirements. (i) Heterogeneous and multi-relational semantics require type- and relation-aware propagation, as well as compositional reasoning over typed multi-hop patterns. (ii) Higher-order and structural signals call for representations that capture mesoscopic substructures, structural roles, and multi-way interactions beyond pairwise edges. (iii) Temporal and dynamic effects demand time-aware embeddings that can encode irregular event timing and retain long-range history under efficiency constraints. Accordingly, this section reviews GRL for complex network structures along three dimensions: (i) heterogeneous and multi-relational networks, (ii) higher-order and structural representations, and (iii) temporal and dynamic graph learning. Each dimension relaxes a distinct classical assumption and, collectively, they form a practical taxonomy for modern complex-network representation learning. A schematic overview of the three-dimensional taxonomy is provided in Figure 7, linking heterogeneous, higher-order, and temporal settings to their modeling requirements.

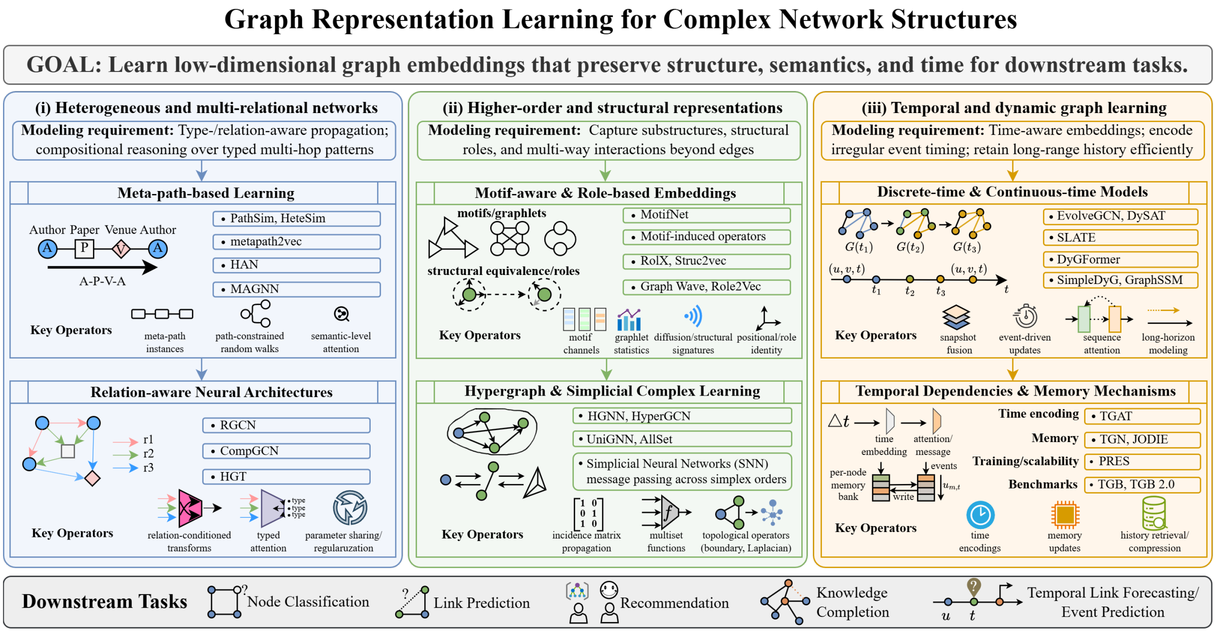

Figure 7. An overview of graph representation learning for complex network structures.

4.1. Heterogeneous and Multi-Relational Networks

Heterogeneous and multi-relational graphs are prevalent in knowledge graphs, biomedical interaction networks, and social systems, where nodes belong to multiple semantic types and edges encode relation-specific meanings [82,83]. The goal of heterogeneous graph representation learning is to learn embeddings that preserve type-aware structural proximity and relation semantics, so that downstream tasks such as node classification, link prediction, recommendation, and knowledge base completion can be performed under the correct semantic context.

Methodologically, heterogeneous representation learning must account for two core aspects: (i) relation-specific semantics, where different edge types correspond to different interaction mechanisms and thus should be modeled with distinct inductive biases; and (ii) compositional semantics, where meaningful associations are often expressed by typed multi-hop patterns rather than single edges. Existing methods can be summarized into two mainstream paradigms: (i) meta-path-based learning and (ii) relation-aware neural architectures. The former emphasizes explicit, interpretable semantic channels through typed relational compositions, while the latter emphasizes end-to-end learning via relation-conditioned propagation directly on the observed multi-relational graph.

4.1.1. Meta-Path-Based Learning

Meta-path-based learning encodes heterogeneous semantics via explicit and interpretable typed patterns. The central idea is to define a set of schema-consistent relational compositions (meta-paths) that represent task-relevant semantics, and then learn embeddings by measuring proximity or aggregating information along instances of these patterns [84].

A foundational direction models typed proximity through meta-path-based similarity and relevance. PathSim quantifies similarity between same-type nodes by counting and normalizing path instances under a symmetric meta-path, producing an interpretable measure of semantic proximity [85]. HeteSim generalizes this principle to relatedness between arbitrary node types under potentially asymmetric meta-paths, providing a unified framework for path-constrained relevance measurement [86]. These works establish meta-paths as explicit semantic operators on heterogeneous schemas.

To improve scalability and generalization, another line uses meta-path-guided random walks with distributional objectives. Metapath2vec performs random walks constrained by predefined meta-paths and optimizes a skip-gram objective so that embeddings preserve meta-path-conditioned neighborhoods [87]. While effective and efficient, this family typically relies on the quality and coverage of the chosen meta-path set.

More recent approaches integrate meta-path semantics into neural aggregation with learnable weighting across channels. HAN employs hierarchical attention that aggregates neighbors under each meta-path and then learns semantic-level attention to combine multiple meta-path channels, enabling end-to-end optimization while retaining interpretability [37]. MAGNN further refines intra-meta-path aggregation by explicitly modeling the contribution of intermediate nodes before fusing multiple meta-path-based representations [88]. Overall, meta-path-based learning provides controllable semantic bias and strong interpretability, but can be sensitive to manual meta-path design and may require additional mechanisms to handle large or evolving schemas.

4.1.2. Relation-Aware Neural Architectures

Relation-aware neural architectures learn heterogeneous representations without explicit meta-path enumeration by defining relation-conditioned message passing on the multi-relational edge set. The key objective is to let each relation type induce an appropriate propagation operator, so that heterogeneous semantics are disentangled through learned transformations, attention, or gating.

Relational graph convolution is a representative instantiation. R-GCN assigns relation-specific transformations in neighborhood aggregation and introduces parameter-sharing strategies to control complexity in graphs with many relations, making it a widely used backbone for multi-relational learning and knowledge base completion [89]. Building on this idea, CompGCN jointly embeds entities and relations and composes them to form relation-aware messages, improving expressiveness while remaining parameter-efficient in multi-relational settings [90].

Attention-based designs offer a more flexible mechanism for heterogeneous propagation. HGT generalizes transformer-style attention to heterogeneous graphs by introducing node-type- and edge-type-dependent projection and attention parameters, enabling scalable learning over complex schemas [91]. In practice, relation-aware architectures are attractive when relation types are numerous, directional, or sparse, since they avoid enumerating discrete semantic paths and instead learn relation-specific operators directly from data. However, effective deployment often depends on careful parameter sharing and regularization to prevent over-parameterization in large relation spaces.

4.2. Higher-Order and Structural Representations

Higher-order and structural representations target mesoscopic organization and interaction patterns that are not fully explained by pairwise edges or 1-hop neighborhoods alone. In many real-world networks, predictive cues reside in recurring substructures (e.g., motifs, graphlets, anchored subgraphs), community-level wiring patterns, and functional roles that reflect structural equivalence rather than proximity. Accordingly, the goal is to learn embeddings that preserve both (i) recurring higher-order patterns and (ii) structural roles/positional functions, enabling downstream tasks to leverage structure-driven semantics beyond local adjacency.

From a modeling perspective, prior work can be organized into two complementary directions. (i) Motif-aware and role-based embeddings inject explicit higher-order structural bias while still operating on the standard graph abstraction. (ii) Hypergraph and simplicial complex learning instead generalizes the relational domain to represent multi-way interactions and higher-dimensional topology as first-class objects, supporting native propagation over set- or simplex-level relations.

4.2.1. Motif-Aware and Role-Based Embeddings

Motif-aware and role-based learning shift the representation focus from ”edge-level proximity” to ”structure-level semantics”, where signal is determined by a node’s participation in higher-order patterns and its structural identity in the network. Motif-aware representations characterize recurrent higher-order building blocks that appear disproportionately often in complex systems and are linked to system-level functions [92,93]. Complementarily, role-based representations emphasize structural equivalence: nodes can exhibit similar functional behavior even when topologically distant because they occupy analogous positions in the global organization of the graph [94].

Motif-aware methods encode how nodes participate in recurring local structures by incorporating motif- or graphlet-level operators into learning. A common design constructs motif-induced operators (e.g., motif adjacency, motif Laplacians, or graphlet-based statistics) and performs motif-conditioned aggregation, followed by attention or gating to fuse multiple motif channels [95–98]. Recent work further integrates motif priors into self-supervised and geometric learning regimes, including motif-aware Riemannian representation learning with generative–contrastive objectives [99], motif-aware masking for molecular graph pretraining [100], and motif-driven contrastive frameworks using motif prototypes [101].

Role-based methods preserve structural identities and positional functions rather than homophily-driven neighborhood similarity. Early work operationalizes roles via feature-based role discovery and factorization, exemplified by RolX [94]. Subsequent approaches compare structural signatures across multi-scale neighborhoods, such as struc2vec [102] and diffusion-based structural embeddings like GraphWave [103], as well as role-oriented random-walk generalizations such as Role2Vec [104]. Recent work revisits scalability and flexibility for structural identity preservation (e.g., ffstruc2vec) [105]. In parallel, graph Transformers have reinforced the importance of explicit positional and structural signals: the discriminative power of graph Transformers is strongly governed by structural encodings [106], and transferable encoders have been proposed to learn general-purpose positional and structural representations to augment downstream GNNs [107].

Overall, motif-aware embeddings contribute an explicit inductive bias toward higher-order local patterns, whereas role-based embeddings contribute a bias toward global structural-function identity. In complex networks where both local participation and global roles are predictive, integrating motif-conditioned aggregation with role/position-aware signals provides a principled path to richer and more stable representations.

4.2.2. Hypergraph and Simplicial Complex Learning

Hypergraph and simplicial complex learning targets relational settings in which interactions are inherently multi-way (e.g., group events, team collaborations, molecular complexes) [108]. Compressing such observations into pairwise edges can distort dependency structure and attenuate group-level semantics. By treating set-level relations (hyperedges) and higher-dimensional simplices as first-class modeling objects, these approaches enable principled propagation over multi-way interactions and support representations faithful to the data-generating mechanism.

In hypergraph learning, most neural architectures propagate information through the incidence structure, typically following a node→hyperedge→node scheme with learnable weighting and normalization. Representative instantiations include HGNN [109] and HyperGCN [110]. To improve portability and connect hypergraph learning with message passing on graphs, UniGNN provides a unified formulation that adapts mainstream GNN operators to hypergraphs [111], while AllSet frames hypergraph layers as compositions of multiset functions [112]. Recent work further explores attention-based designs tailored to hypergraphs, such as topology-guided hypergraph Transformers with structural encodings [113]. Surveys systematically organize the design space and evaluation practices [30,114], and one-stage message passing has been proposed to better reconcile local aggregation with global mixing effects [115].

Simplicial complexes generalize graphs by modeling higher-order simplices together with boundary and co-boundary operators, allowing signals to be propagated across simplex orders. Simplicial Neural Networks introduce convolutional operators grounded in simplicial topology [116]. More recent frameworks couple representations across 0/1/2-simplices via structured message passing [117], and attention-based extensions employ masked self-attention aligned with simplicial neighborhoods and topological operators [118]. Tooling efforts such as TopoX support reproducible computation and learning on topological domains beyond graphs [119].

Overall, hypergraph and simplicial learning are particularly advantageous when the underlying mechanism is genuinely multi-way or topological. Compared with motif-based augmentation on graphs, these approaches promote higher-order relations to primary modeling entities, albeit at the cost of more complex operators, additional modeling choices, and potentially higher computational overhead.

4.3. Temporal and Dynamic Graph Learning

Temporal and dynamic graph learning concerns systems whose topology and interactions evolve over time: interactions are timestamped, edges may emerge or vanish, and node states can drift as new evidence arrives. The objective is to learn time-aware embeddings that capture both structural context and temporal dynamics, supporting temporally grounded tasks such as forecasting future links, predicting events, and performing dynamic node classification under chronological constraints.

A useful taxonomy distinguishes two complementary perspectives: (i) discrete-time vs. continuous-time formulations, which contrast snapshot sequences with event-driven interaction streams; and (ii) temporal dependencies and memory mechanisms, which focus on how models encode time gaps, retrieve relevant history, and compress long interaction traces under efficiency constraints.

4.3.1. Discrete-Time and Continuous-time Models

Discrete-time approaches model a dynamic graph as a sequence of snapshots and couple representations across time windows via recurrent evolution, self-attentive temporal fusion, or temporal regularization. This formulation is conceptually clear and compatible with standard message passing, but it often coarsens fine-grained temporal information, since event order and irregular inter-event intervals within a snapshot are not explicitly preserved. Representative snapshot-based architectures include EvolveGCN [120] and DySAT [121]. Recent discrete-time Transformer developments incorporate structured spatio-temporal encodings; for example, SLATE leverages a supra-Laplacian construction to encode discrete-time dynamic graphs as multi-layer structures and couples this encoding with Transformer-style modeling for dynamic link prediction [122].

Continuous-time approaches treat interactions as a timestamped event stream and update node states upon events, preserving temporal order and accommodating irregular time gaps. Transformer-style designs have become influential because attention over historical interaction sequences can capture long-range temporal effects while flexibly integrating time encodings and history summarization. DyGFormer learns from historical first-hop interaction sequences via neighbor co-occurrence encoding and sequence patching [123]. Complementary evidence suggests that well-designed, comparatively simple sequence models can remain highly competitive on temporal link prediction [124], motivating careful assessment of architectural complexity. SimpleDyG explores a minimalist Transformer design with temporal alignment [125]. Beyond attention, state-space formulations have been introduced for long-horizon modeling at scale; GraphSSM extends state-space model theory to temporal graphs via Laplacian-regularized objectives [126]. Recent work also revisits temporal-Transformer design from a time-series perspective to better capture periodic patterns and long-term dependencies [127].

Evaluation studies have shown that performance can be strongly affected by negative sampling choices and edge recurrence, and proposed memorization-based baselines to expose overly permissive settings [128]. Building on these insights, the Temporal Graph Benchmark (TGB) standardizes datasets, protocols, and evaluation pipelines with an emphasis on realistic forecasting constraints and reproducibility [129]. TGB 2.0 extends this effort to temporal knowledge graphs and temporal heterogeneous graphs, highlighting the practical importance of relation types, scalability, and competitive heuristic baselines [130]. Surveys synthesize these developments and organize temporal interaction graph representation learning into coherent taxonomies [131].

4.3.2. Temporal Dependencies and Memory Mechanisms

Temporal graphs exhibit multi-scale dependencies under irregular event timing, where predictive signals may arise from short-term recency, medium-range periodicity, and long-horizon influence accumulation. Capturing such heterogeneity typically relies on two complementary components: (i) time encoding mechanisms that parameterize time gaps and temporal order, and (ii) history and memory modeling mechanisms that store, retrieve, and compress interaction histories so that long-range dependencies remain accessible under efficiency constraints.

Time encodings provide explicit inductive bias for representing irregular inter-event intervals and temporal order in a form consumable by message passing or attention. Canonical designs include functional or basis-based encodings that modulate temporal aggregation, as exemplified by TGAT [132]. More recent work explores richer spatio-temporal positional signals for continuous-time dynamic graphs, including correlated spatial-temporal positional encodings [133] and learnable spatial-temporal positional encodings for link forecasting [134]. Transformer formulations can also incorporate continuous-time dynamics through dedicated time-encoding modules [135].

Complementary to time encoding, memory-centric designs maintain persistent node-wise states that evolve upon events, enabling models to retain information across sparse interactions and support long-horizon reasoning. Temporal Graph Networks (TGN) provide a general event-driven framework for temporal node embeddings with explicit memory updates [136]. Earlier models such as JODIE additionally emphasize forecasting embeddings into the future via learned projection operators [137]. Other paradigms retrieve history through temporally constrained walks, where anonymized walk statistics summarize temporal motifs and local evolution patterns [138]. Recent advances increasingly target scalability and training stability of memory-based temporal GNNs; for example, PRES studies efficient training under large temporal batches and addresses temporal discontinuity in parallel processing [139]. Surveys consolidate these mechanisms and clarify design trade-offs among time encoding, memory updates, and history compression [140,141].

5. Learning Paradigms and Optimization Strategies

GRL has experienced a substantial evolution over the past decade, shifting from purely label-dependent supervised approaches toward more versatile, data-efficient, and robust frameworks [18,142]. This shift has been largely motivated by challenges inherent to complex networks, including the limited availability of high-quality annotations and the necessity for learned representations to remain effective across heterogeneous structures, noisy observations, and distributional variations [28,143]. Contemporary methods address these challenges not merely through novel architectures, but also through distinct learning paradigms and corresponding optimization objectives that govern how information is extracted and encoded from graph-structured data. In particular, the learning paradigm refers to a category of methods with shared principles, whereas the corresponding optimization objective denotes the specific training target that guides the learning process within that paradigm. This distinction is applied consistently throughout the following description.

In this section, we present a structured overview of GRL, categorizing existing methodologies into four main paradigms: supervised and semi-supervised learning, self-supervised and contrastive learning, generative graph representation learning, and causal graph learning. For each paradigm, we examine the underlying learning mechanisms, the specific optimization targets they pursue, the inductive biases they introduce, and the strengths and limitations that arise when applying these methods to real-world complex networks.

5.1. Supervised and Semi-Supervised Learning

Supervised and semi-supervised learning constitute the most classical learning paradigms in graph representation learning [60,144]. Both paradigms rely on explicit ground-truth labels as the primary source of supervision and optimize label-driven empirical risk objectives. Their fundamental difference lies in whether and how unlabeled graph entities are incorporated into the optimization process. In the following, we first introduce these two paradigms from a unified optimization perspective and then analyze their respective strengths and limitations in complex network settings.

5.1.1. Supervised Learning

In supervised GRL, a graph $\mathcal{G} = (\mathrm{V}, \mathrm{E}, \mathbf{X})$ is partially or fully annotated with ground-truth labels defined on nodes, edges, or the entire graph. Let $\mathrm{V}_l \subseteq \mathrm{V}$ denote the set of labeled nodes, and let y_i denote the corresponding label of node v_i. The objective of supervised learning is to learn a node representation function $\mathcal{F}_{\text{node}}(\cdot)$ and a task-specific predictor $f(\cdot)$ by minimizing the supervised loss $\mathcal{L}_{\text{sup}}$ over labeled nodes:

$$\mathcal{L}_{\text{sup}}(\Theta) = \frac{1}{|\mathcal{V}_l|} \sum_{v_i \in \mathcal{V}_l} \ell\big(f(\mathbf{h}_i), y_i\big), \quad \mathbf{h}_i = \mathcal{F}_{\text{node}}(v_i \mid \mathcal{G}; \Theta), \tag{20}$$

where Θ denotes all learnable parameters of both the representation function and the predictor, $\ell(\cdot)$ is a supervised loss function such as cross-entropy.

From an optimization perspective, supervised learning assumes that labeled samples are sufficiently abundant, accurate, and representative of the underlying data distribution [145]. Under these assumptions, the learned representations are directly optimized toward the downstream task objective. However, in real-world complex networks, labels are often sparse, noisy, and unevenly distributed, which leads to biased empirical risk minimization and poor generalization, particularly in large-scale or dynamically evolving graphs.

5.1.2. Semi-Supervised Learning

Semi-supervised learning extends the supervised paradigm by incorporating unlabeled nodes $\mathrm{V}_u = \mathrm{V} \setminus \mathrm{V}_l$ into the training process. Rather than redefining the supervision source, semi-supervised methods retain the label-driven objective as the core optimization target, while introducing auxiliary regularization terms that exploit structural or distributional properties of unlabeled data. A generic semi-supervised objective can be expressed as:

$$\min_{\Theta} \ \mathcal{L}_{\text{semi}} = \mathcal{L}_{\text{sup}} + \lambda \mathcal{R}(\mathbf{H}, \mathcal{G}), \tag{21}$$

where $\mathcal{R}(\cdot)$ denotes a regularization term defined on the learned representations $\mathbf{H}$ and the graph structure $\mathcal{G}$, and λ controls the trade-off between supervised loss and auxiliary regularization.

From a paradigm-level viewpoint, semi-supervised learning does not fundamentally alter the optimization goal of supervised learning; instead, it leverages unlabeled data to stabilize training and improve generalization [146,147]. This process typically relies on additional inductive biases, encoded in the regularization term $\mathcal{R}(\cdot)$, which enforce smoothness or label consistency across neighboring nodes. However, the effectiveness of these auxiliary assumptions is not guaranteed in all network settings. In complex networks with high structural heterogeneity, ambiguous boundaries, or weak homophily, these assumptions may be violated, potentially leading to limited or even negative performance gains.

The above discussion highlights that supervised and semi-supervised learning share a common label-centric optimization core, which enables direct task alignment when reliable annotations are available. Supervised learning offers conceptual simplicity and clear optimization objectives, while semi-supervised learning improves data efficiency by exploiting unlabeled nodes as auxiliary information. Despite their foundational role in graph representation learning, both paradigms are constrained by their dependence on labeled data and the generic structural assumptions often encoded in semi-supervised approaches, which can limit scalability and performance in complex networks. These observations motivate the consideration of alternative learning paradigms that redefine supervision signals and optimization strategies, as will be explored in the following sections.

5.2. Self-Supervised and Contrastive Learning

In scenarios within complex networks where labeled data is scarce or unavailable, self-supervised learning has emerged as a powerful paradigm in GRL [148]. Unlike supervised and semi-supervised learning, these methods do not rely on explicit ground-truth labels. Instead, they design auxiliary pretext tasks or construct contrastive objectives that generate supervisory signals from the data itself [149]. By doing so, these approaches aim to learn expressive node, subgraph, or graph-level representations that capture intrinsic structural and semantic properties, which can later be transferred to downstream tasks.

5.2.1. Self-Supervised Learning

To implement self-supervised learning in graphs, graph self-supervised methods typically define pretext tasks that generate pseudo-labels based on inherent graph properties [150,151]. These tasks can be broadly classified into predictive and generative schemes:

- Predictive Tasks: These objectives enforce the model to predict certain graph attributes or structural features from partial observations. For instance, node feature masking, edge prediction, or context prediction tasks encourage the encoder to capture both local neighborhood patterns and global structural dependencies. Formally, a general predictive objective can be written as

$$\min_{\Theta} \mathcal{L}_{\text{pred}} = \frac{1}{|\mathrm{S}|} \sum_{(i,j)\in \mathrm{S}} \ell_{\text{pred}}\big(s(\mathbf{h}_i, \mathbf{h}_j), y_{ij}^{\text{pseudo}}\big), \tag{22}$$

where S denotes the set of node (or subgraph) pairs involved in the pretext task, $s(\mathbf{h}_i, \mathbf{h}_j)$ is a similarity or scoring function between representations, y_{ij}^{pseudo} is the pseudo-label derived from the graph (e.g., existence of an edge, masked attribute), and $\ell_{\text{pred}}(\cdot)$ is a task-specific loss.

- Generative Tasks: Generative self-supervised objectives aim to reconstruct certain parts of the graph from latent representations. For example, a masked node feature or subgraph reconstruction loss encourages the encoder to capture deeper semantic dependencies rather than relying solely on local adjacency. The objective can be expressed as

$$\min_{\Theta} \mathcal{L}_{\text{gen}} = \frac{1}{|\mathrm{V}|} \sum_{v_i\in \mathrm{V}} \ell_{\text{rec}}\big(\hat{\mathbf{x}}_i, \mathbf{x}_i\big), \tag{23}$$

where $\hat{\mathbf{x}}_i$ is the reconstructed feature of node v_i, and $\ell_{\text{rec}}(\cdot)$ measures reconstruction error, e.g., mean squared error for continuous features.

These pretext objectives allow the model to leverage unlabeled data in a task-agnostic manner, guiding the encoder to capture intrinsic structural and semantic patterns and producing representations that are more generalizable and transferable across a variety of downstream tasks [152,153].

5.2.2. Contrastive Learning

Contrastive learning has become one of the most influential paradigms for GRL, focusing on maximizing agreement between related views while distinguishing unrelated ones [154,155]. Let $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ be the two node representation matrices obtained from two augmented views of the graph, a typical contrastive objective can be written as an InfoNCE loss:

$$\min_{\Theta} \mathcal{L}_{\text{con}} = -\sum_{i=1}^{n_{\mathrm{V}}} \log \frac{\exp(sim(\mathbf{h}_i^{(1)}, \mathbf{h}_i^{(2)})/\tau)}{\sum_{j=1}^{n_{\mathrm{V}}} \exp(sim(\mathbf{h}_i^{(1)}, \mathbf{h}_j^{(2)})/\tau)}, \tag{24}$$

where $sim(\cdot,\cdot)$ is a similarity function (e.g., cosine similarity) and τ is a temperature parameter. This formulation encourages alignment between positive pairs while repelling negative samples, thereby enhancing the discriminative power of the learned embeddings [156].

Graph contrastive learning methods differ in how views are constructed (e.g., node/edge dropping, subgraph sampling, or learned augmentations), the level of comparison (node-node, node-graph, graph-graph), and whether negative samples are explicitly required. Recent advances include asymmetric architectures and non-contrastive objectives, which avoid negative sampling while still preventing representation collapse [157].

Overall, self-supervised and contrastive learning extend the classical label-centric paradigm by generating supervisory signals directly from the graph structure and features. Predictive and generative pretext tasks enable task-agnostic representation learning, while contrastive objectives encourage alignment across multiple graph views.

These approaches reduce reliance on labeled data, improve robustness to noisy or sparse annotations, and have been shown to achieve strong transferability across downstream tasks. Nevertheless, their effectiveness depends on the choice of pretext task, augmentation strategy, and contrastive objective, which can vary significantly across different types of complex networks.

5.3. Generative Graph Representation Learning

In complex networks where capturing the underlying distribution of graph structures or attributes is essential, generative graph representation learning has emerged as a prominent paradigm. Unlike supervised or contrastive methods that primarily rely on labels or pairwise agreement, generative approaches aim to reconstruct graph components or model the probability distribution governing node features and graph connectivity [158, 159]. By doing so, these methods facilitate deeper structural understanding and can support tasks such as link prediction, graph completion, or synthetic graph generation [159].

5.3.1. Graph Autoencoders

Graph autoencoders (GAEs) constitute the most widely adopted generative framework in GRL [160, 161]. A typical GAE comprises an encoder $\mathcal{F}_{\text{node}}(\cdot)$ that maps nodes to latent embeddings $\mathbf{H} \in \mathbb{R}^{n_{\mathrm{V}} \times d_h}$ and a decoder $g(\cdot)$ that reconstructs graph properties from these embeddings. The learning objective can be formalized as

$$\min_{\Theta} \mathcal{L}_{\text{gae}} = \frac{1}{n_{\mathrm{E}}} \sum_{(i,j)\in \mathrm{E}} \ell_{\text{rec}}\big(\hat{a}_{ij}, a_{ij}\big) + \frac{1}{n_{\mathrm{V}}} \sum_{v_i \in \mathrm{V}} \ell_{\text{feat}}\big(\hat{\mathbf{x}}_i, \mathbf{x}_i\big), \tag{25}$$

where $\hat{a}_{ij} \in \hat{\mathbf{A}}$ and $\hat{\mathbf{x}}_i \in \hat{\mathbf{X}}$, $\hat{\mathbf{A}}$ and $\hat{\mathbf{X}}$ are the reconstructed adjacency and feature matrices, $\ell_{\text{rec}}(\cdot)$ and $\ell_{\text{feat}}(\cdot)$ denote reconstruction losses, and Θ encompasses all learnable parameters of the encoder and decoder. This encourages embeddings to capture essential structural and feature information, providing a compact and informative latent representation for downstream tasks [162, 163].

5.3.2. Masked Graph Autoencoders

Inspired by advances in masked language and image modeling, masked graph autoencoders (MGAEs) shift the focus from global reconstruction to contextual recovery [164, 165]. A subset of node features or edges is masked, and the model is trained to predict them from the unmasked context:

$$\min_{\Theta} \mathcal{L}_{\text{mgae}} = \frac{1}{|\mathrm{M}|} \sum_{v_i \in \mathrm{M}} \ell_{\text{rec}}\big(\hat{\mathbf{x}}_i, \mathbf{x}_i\big), \tag{26}$$

where $\mathrm{M} \subseteq \mathrm{V}$ is the set of masked nodes. This strategy encourages the model to capture higher-order dependencies and semantic patterns beyond local adjacency, making representations more robust and informative [166, 167].

Overall, generative graph representation learning provides a principled way to model both structure and attributes, supporting diverse downstream applications that require robust, distribution-aware representations in complex networks. However, its effectiveness depends on the encoder–decoder design, reconstruction targets, and the choice between deterministic and probabilistic modeling, which can influence both scalability and expressive power in large-scale heterogeneous networks.

5.4. Causal Graph Learning

In real-world networks, observed correlations can be confounded by spurious dependencies, distribution shifts, or latent factors, limiting the reliability of purely statistical GRL. Causal graph learning (CGL) addresses these challenges by integrating causal reasoning into representation learning, aiming to disentangle genuine structural effects from spurious correlations [168, 169].

5.4.1. Causal Representation Learning

CGL frameworks often seek to decompose node or subgraph representations into invariant causal components and environment-dependent factors [170]. Formally, let $\mathbf{h}_i = \mathcal{F}_{\text{node}}(v_i \mid \mathcal{G}; \Theta)$ be the embedding of node v_i under a set of observed environments $\mathcal{E}$. Causal representation learning enforces invariance across these environments:

$$\min_{\Theta} \mathcal{L}_{\text{crl}} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \ell_{\text{sup}}\big(f(\mathbf{h}_i^e), y_i\big) + \gamma\, \mathcal{R}_{\text{inv}}(\mathbf{H}^e), \tag{27}$$

where ℓ_{sup} is a supervised or pseudo-supervised loss, $\mathcal{R}_{\text{inv}}(\cdot)$ encourages invariance across environments, and γ balances the two terms. This formulation enhances out-of-distribution generalization by encouraging the learned representations to capture invariant causal mechanisms, and reduces reliance on spurious correlations that may arise from confounding or dataset-specific biases [171].

5.4.2. Causal Reasoning and Interventions

Beyond invariant representation, CGL also leverages interventions and counterfactual reasoning to estimate causal effects of specific nodes, edges, or subgraphs [172, 173]. For example, given a subgraph $\mathcal{G}_{sub} \subseteq \mathcal{G}$, an intervention such as removing an edge $e_{ij} \in \mathcal{G}_{sub}$ allows the model to learn how this change affects predictions for nodes within $\mathcal{G}_{sub}$:

$$\hat{y}_i^{do(e_{ij}=0)} = f\big(\mathcal{F}_{\text{node}}(\mathcal{G}_{sub} \setminus e_{ij})\big), \quad v_i \in \mathcal{G}_{sub}, \tag{28}$$

where $do(e_{ij} = 0)$ denotes a causal intervention that sets the edge e_{ij} to be absent, following the do-calculus formalism in causal inference. This approach provides a principled understanding of cause-and-effect relationships within the subgraph, facilitating the quantification of node and edge influence, identification of critical substructures, and interpretable insights into model predictions beyond mere correlations [174].

In summary, CGL extends GRL beyond mere correlations by incorporating invariant representations and interventional reasoning, providing more trustworthy and generalizable embeddings for downstream tasks. Causal modules can improve fairness by mitigating bias along sensitive paths, enhance interpretability by identifying the true causal substructures driving predictions, and strengthen robustness against distribution shifts. Nevertheless, designing effective causal objectives, obtaining suitable interventional data, and identifying appropriate environments remain challenging, particularly in networks with heterogeneous connectivity and evolving structures.

6. Applications In Real-World Complex Networks

As a universal paradigm for modeling real-world systems, complex networks provide a novel perspective for understanding and processing relational data. Currently, this theory transcends purely theoretical frameworks, permeating widely into practical applications across multiple disciplines and achieving significant progress. Although application scenarios are increasingly diverse, existing research mainly focuses on the following four key areas: In the realm of biological and biomedical networks, advanced graph representation learning provides powerful tools for elucidating complex association patterns within multi-scale biological systems, thereby laying a data-intelligence foundation for precision medicine [175–180]. In social and information networks, graph-based algorithms constitute the technical backbone for capturing dynamic interaction behaviors and constructing mechanisms for personalized information dissemination [181–184]. Regarding traffic, energy, and infrastructure networks, network modeling oriented toward spatiotemporal dependencies reshapes the capability to optimize resource allocation and enhance system resilience [185, 186]. The deep integration of knowledge graphs with industrial systems redefines the boundaries of intelligent manufacturing and automated reasoning by empowering machines to comprehend complex semantic relationships [187,188]. These cross-domain advancements collectively indicate that, despite the heterogeneity of data modalities, the core challenge remains how to effectively represent and mine latent complex structural patterns.

6.1. Biological and Biomedical Networks

The intrinsic complexity of biological systems lies in the fact that they are not merely simple collections of isolated molecules, but rather dynamic networks constituted by intricate interactions among genes, proteins, and metabolites [189–191]. With the explosive growth of high-throughput multi-omics technologies, network biology transitions from early descriptive statistical analysis toward a new era of artificial intelligence-based predictive modeling. As Barabási et al.[192] point out in their seminal work on network medicine, diseases often do not arise from single gene mutations but rather result from perturbations within specific network modules. However, in the face of exponentially growing multi-omics data, traditional data integration methods confront severe challenges, including high dimensionality, sparsity, and heterogeneity [193].

To effectively decode these complex biological maps, the evolution of computational paradigms in biological network analysis can be viewed as a shift from early graph-theoretical approaches, which predominantly relied on manually crafted features like degree and betweenness centrality, toward modern representation learning techniques, such as Graph Neural Networks (GNNs) and network embedding. Early methods were often constrained by their reliance on predefined structural features, which limited their ability to capture the complex, dynamic relationships inherent in biological systems. In contrast, recent advancements leverage data-driven models to automatically learn more nuanced representations, enabling a more comprehensive understanding of biological interactions and improving the predictive power of the models. Zitnik et al. [194] emphasize that machine learning, particularly deep learning, emerges as a critical tool for integrating multimodal biomedical data to infer new knowledge. Review studies by Yue et al. [195] and Yi et al. [196] further demonstrate that these emerging deep learning architectures automatically extract non-linear topological features and rich biological semantics, significantly enhancing performance in downstream analysis tasks. Furthermore, Zhang et al. [197] point out that current algorithmic innovations dedicate efforts to addressing the dynamics and interpretability of graph data. Based on this, this review categorizes the core research directions of biological network analysis into three key classes: identification of key biomedical entities (node level), functional module mining and clustering (subgraph level), and link prediction and relationship inference (edge level).

The evolution of computational paradigms in biological network analysis can be viewed as a shift from early graph-theoretical approaches, which predominantly relied on manually crafted features like degree and betweenness centrality, toward modern representation learning techniques, such as Graph Neural Networks (GNNs) and network embedding. Early methods were often constrained by their reliance on predefined structural features, which limited their ability to capture the complex, dynamic relationships inherent in biological systems. In contrast, recent advancements leverage data-driven models to automatically learn more nuanced representations, enabling a more comprehensive understanding of biological interactions and improving the predictive power of the models.

6.1.1. Identification of Key Biomedical Entities

In the frontier exploration of biological network analysis, the identification of Critical Nodes serves as a core task for understanding the robustness and functional regulation of complex systems [198,199]. As research paradigms evolve, the focus of this field shifts profoundly from simple topological structure analysis to the utilization of Graph Foundation Models and Large Language Models (LLMs) to precisely pinpoint entities that play decisive roles in biological processes. To address the limitations of traditional methods in identifying potential therapeutic targets, Huang et al. [200] develop TxGNN, a model that integrates large-scale medical knowledge graphs and introduces zero-shot prediction mechanisms, successfully achieving precise identification and repurposing of critical drug target nodes within rare disease networks. In the realm of designing critical functional nodes for protein interaction networks, the Chroma model proposed by Ingraham et al. represents a major breakthrough in generative AI; this work utilizes diffusion models combined with geometric graph constraints to not only identify existing critical domains but also design protein nodes with specific functions de novo, thereby reshaping the paradigm for screening and designing critical nodes [201]. Addressing regulatory networks at the single-cell level, Cui et al. [202] develop scGPT, which transfers generative pre-trained Transformers to transcriptomic data; by constructing a foundation model for cell atlases, it achieves high-precision identification of critical gene nodes that determine cell fate specificity. Furthermore, to locate critical entities in complex networks where labels are scarce, Liu et al. propose MoleculeSTM, which innovatively combines textual descriptions with molecular graph structures; utilizing multi-modal contrastive learning strategies, the model directly identifies critical molecular nodes with specific chemical attributes based on natural language instructions, marking the entry of critical node identification technology into a new stage driven by ”text-graph” interaction [203].

6.1.2. Functional Module Mining and Clustering

Functional module mining aims to disentangle cooperative molecular machinery from complex biological networks [204, 205]. Its core challenge lies in effectively integrating high-dimensional multi-omics data while overcoming the inherent heterogeneity and dynamics of biological systems. Current algorithmic paradigms have moved well beyond traditional single-network community detection, shifting comprehensively toward deep graph clustering and multi-view contrastive learning.

At the molecular and single-cell levels, recent studies increasingly adopt deep learning-based graph clustering frameworks to model complex biological interactions. Li et al. propose M6A-DCR, a deep learning model that mines consensus regions via graph contrastive clustering, achieving high-precision identification of RNA m6A modification sites while providing interpretability at the motif level [198]. Wang et al. present the single-cell graph neural network (scGNN), a hypothesis-free deep learning framework combining graph neural networks with a left-truncated mixture Gaussian model. By iteratively integrating gene expression matrices and cell-cell interaction graphs through a multi-modal autoencoder, this framework effectively addresses the sparsity of scRNA-Seq data and outperforms existing methods across multiple benchmarks. It not only improves the accuracy of cell clustering and gene imputation but also successfully applies to Alzheimer’s disease research, revealing novel disease-related mechanisms [206]. Beyond deep clustering paradigms, emerging studies further emphasize cross-modal fusion and high-order structural reasoning. Su et al. develop an interpretable Transformer model fusing multi-omics data with complex biological network topology, breaking through the bottlenecks of generalization and interpretability in existing graph learning methods to achieve precise cancer gene prediction across various cancer scenarios. This model not only identifies potential novel oncogenes but also serves as a powerful tool for revealing deep gene regulatory mechanisms and mining key functional modules by parsing high-order structural features [207]. At the tissue and spatial omics scale, functional module mining further extends toward large-scale spatial graph modeling. The BANKSY algorithm innovatively unifies cell typing and tissue domain segmentation tasks in spatial omics by fusing the transcriptomic features of cells with their local neighborhoods. This framework surpasses existing methods in accuracy, speed, and scalability, efficiently handling million-cell datasets and addressing quality control and batch effect issues [208].

In summary, functional module mining is evolving deeply toward high-order feature fusion and cross-modal synergy; future research focuses more on achieving cross-scale analysis—from microscopic molecular interactions to macroscopic tissue structures—within a unified computational framework, thereby precisely localizing dynamic modules with specific biological functions.

6.1.3. Link Prediction and Relationship Inference

Link prediction and relationship inference serve as core mechanisms for elucidating latent associations within complex biological systems [209–212]. As research deepens, this field transcends the limitations of early approaches based on simple similarity metrics, shifting towards utilizing deep learning frameworks to mine deep features within Heterogeneous Information Networks (HINs) and knowledge graphs. Current cutting-edge methods demonstrate that this domain undergoes a profound paradigm shift: evolving from pattern recognition relying solely on topological structures to an advanced stage that integrates multi-modal data, strengthens semantic reasoning, and incorporates contrastive learning and Transformer architectures.

In addressing complex biological entity relationships, extracting robust structural features from heterogeneous networks remains a primary challenge. Traditional graph learning methods are being superseded by more refined mechanisms. For instance, in Drug-Drug Interaction (DDI) prediction, Spatial-aware Capsule networks are introduced into knowledge graph neural networks, aiming to capture fine-grained structural patterns within the interaction space [213]. Similarly, in the domain of Drug-Disease Association (DDA), HINGRL [214] demonstrates the importance of mining complex network topology for identifying potential associations by applying graph representation learning on Heterogeneous Information Networks. These works establish the foundation for inference based on structural features.

Pure structural information often fails to capture the intricate biochemical mechanisms between biomolecules; therefore, introducing semantic reasoning and long-range dependency modeling becomes key to enhancing prediction performance. In Drug-Target Interaction (DTI) research, Mhtan-dti [215] innovatively utilizes a meta-path-based hierarchical Transformer and attention network, successfully decoding long-range semantic dependencies within the network. Furthermore, LLM-DDI [216] represents the frontier of semantic reasoning; it leverages the powerful text understanding capabilities of Large Language Models (LLMs) to combine biomedical textual knowledge with topological data, significantly enhancing the model's ability to comprehend and predict within biomedical knowledge graphs.

To address the heterogeneity and noise of biological data, multi-modal fusion becomes a mainstream strategy for improving model robustness. By integrating different views and data sources, models acquire more comprehensive entity representations. MMDG-DTI [217] effectively resolves the issue of data heterogeneity in DTI prediction through multi-modal feature fusion and domain generalization techniques. Meanwhile, in DDA prediction, AMDGT [218] adopts an attention-aware multi-modal fusion within a dual-graph Transformer architecture, which further demonstrates the effectiveness of comprehensively utilizing multiple data modalities for enhancing prediction accuracy.

In the face of confusion caused by highly similar entities, contrastive learning strategies demonstrate superior discriminative ability. By pulling positive samples closer and pushing negative samples apart, the model learns more discriminative representations. In DDI prediction, Multi-view Contrastive Learning strategies [219] are employed to align distinct drug representations; whereas in the DDA domain, NCH-DDA [220] implements neighborhood contrastive learning, focusing on distinguishing extremely subtle functional association differences. This strategy significantly enhances the precision of models in high-confidence relationship inference.

In summary, the fields of link prediction and relationship inference are in a stage of rapid development. From improvements in heterogeneous network representation learning to various complex architectures based on Transformers and contrastive learning, these studies collectively outline a clear trend: future biological relationship inference relies more heavily on deep semantic understanding and structural fusion of multi-scale and multi-modal data.

6.2. Social and Information Networks

Research on applications in social and information networks primarily focuses on understanding the underlying topological structures of networks and their influence on node behavior, with community detection and link prediction serving as two cornerstone tasks [221,222]. Early studies mainly concentrate on static graph structures, utilizing modularity optimization or spectral clustering methods to identify dense subgroups within networks [223,224]. With the introduction of deep learning, methods based on Graph Neural Networks (GNNs) greatly enhance the performance of these tasks. For instance, the Graph Convolutional Network (GCN) proposed by Kipf et al. effectively captures local structural features of nodes by aggregating neighbor information, thereby significantly improving the accuracy of semi-supervised classification and link prediction [77]. Furthermore, addressing Heterogeneous Information Networks (HINs), researchers develop models such as metapath2vec, which successfully resolve the semantic embedding issues of multi-typed nodes and edges through meta-path random walk strategies, providing new perspectives for complex social network analysis [87].

Information diffusion and influence maximization represent another core topic in this field, aiming to identify key nodes within a network that maximize information spread. The classical framework for this problem is established by Kempe et al., who formalize it as a discrete optimization problem and demonstrate its NP-hard nature [225]. Recent research begins to integrate temporal dynamics and content features, striving to address efficiency bottlenecks in large-scale networks. For example, sketch-based algorithms significantly reduce computational complexity while maintaining theoretical approximation ratios [226]. Meanwhile, with the proliferation of fake news and rumors on social media, rumor detection becomes an urgent application direction. Modern models not only utilize the structural features of propagation trees but also introduce Transformer-based architectures to capture long-range semantic dependencies between source tweets and retweet comments, thereby effectively curbing the spread of misinformation at an early stage [227,228].

Social network-based recommendation systems represent the forefront of commercial applications for information networks. Traditional collaborative filtering algorithms often face cold-start and data sparsity problems, whereas the introduction of social relationships effectively mitigates these challenges. Models such as GraphRec simultaneously model the user-item interaction graph and the user-user social graph via graph neural networks, utilizing attention mechanisms to distinguish the influence of varying social strengths, thereby achieving more precise personalized recommendations [229]. The latest research trends further explore the application of self-supervised learning in recommendation systems, mining latent views within the graph structure through contrastive learning paradigms to enhance model robustness under sparse data conditions [230]. This hybrid recommendation mode, which integrates social trust with explicit interactions, is gradually becoming the mainstream paradigm for modern information network applications.

6.3. Transportation, Energy, and Infrastructure Networks

Transportation networks serve as the cornerstone of modern urban operations, where network science principles optimize traffic flow and system resilience. Foundational research models these systems as complex networks and analyzes topological properties to understand system vulnerability under random failures or deliberate attacks [231]. Current methodologies integrate deep learning with graph theory to predict traffic congestion in real-time and optimize route planning. For instance, Spatio-Temporal Graph Convolutional Networks (ST-GCNs) capture dynamic correlations among traffic nodes, which significantly improves prediction accuracy compared to traditional time-series models [232]. Recent frontier research extends these frameworks to autonomous vehicle fleets and investigates how connected vehicle networks mitigate traffic bottlenecks through cooperative driving strategies; related review articles discuss the potential impact of connected vehicles on traffic flow stability in detail [233]. Furthermore, researchers emphasize the complexity of multi-modal transportation systems and employ a multilayer network perspective to analyze urban traffic, revealing how the coupling between different transportation layers (such as bus and subway) affects overall system accessibility and efficiency [234].

Energy networks, particularly power grids, undergo a fundamental transformation toward decentralization and intelligence, driven by the integration of renewable energy sources. Classical analysis of power grids focuses on synchronization stability and the prevention of cascading blackouts, treating the grid as a system of coupled oscillators [235]. Modern smart grids leverage Internet of Things (IoT) sensors and Advanced Metering Infrastructure (AMI) to enable bidirectional communication between utility providers and consumers. Recent advancements apply Graph Neural Networks (GNNs) to fault diagnosis and state estimation, allowing operators to locate anomalies in distribution networks with high precision, even under data-scarce conditions [236]. Current research hotspots explore intelligent algorithms in energy management, specifically the application of Reinforcement Learning in demand response, where algorithms automatically balance building energy consumption with grid supply to address the volatility of renewable energy [237]. Furthermore, the security of these cyber-physical systems remains a critical focus, with new detection algorithms emerging to identify false data injection attacks in real-time [238].

Infrastructure networks operate as interdependent systems where the failure of one component often propagates to others, necessitating holistic modeling approaches. The concept of interdependent networks highlights the coupling between distinct infrastructures; for instance, the dependence of water systems on electricity, and vice versa, induces abrupt phase transitions that increase systemic vulnerability [239]. Contemporary research addresses the resilience of critical infrastructure against extreme weather events and climate change. Scholars now employ digital twin technology to create high-fidelity virtual replicas of physical assets, thereby enabling predictive maintenance and scenario planning for aging infrastructure [240]. In recent years, the sustainability of infrastructure networks receives significant attention, with studies utilizing multi-objective optimization algorithms to optimize the lifecycle carbon footprint of network expansion. These integrated frameworks facilitate scientific decision-making, balancing operational efficiency, structural resilience, and environmental sustainability in the planning of next-generation urban infrastructure [241].

6.4. Knowledge Graphs and Industrial Systems

In the era of Industry 4.0, the manufacturing sector faces the challenge of exploding data volumes contrasted with low knowledge utilization rates. Industrial Knowledge Graphs (IKGs), as a semantic technology, transform discrete manufacturing data into interconnected knowledge networks. The application of knowledge graphs in manufacturing extends from simple querying to complex decision support, effectively integrating heterogeneous data sources and breaking down "information silos" within production processes [242]. The construction of such large-scale industrial knowledge graphs is not merely an academic exploration; its effectiveness in handling massive entity relationships is demonstrated by technology giants like Google and Amazon, and it gradually permeates traditional industrial sectors [243].

In specific smart manufacturing scenarios, knowledge graphs are employed to resolve complex issues regarding process planning and resource management. For instance, Bao et al. propose a semantic modeling method for manufacturing resources based on knowledge graphs. This approach enables machines to comprehend the capabilities and limitations of processing equipment, thereby achieving more precise automated process planning [244]. Through these technologies, enterprises convert unstructured expert experience into structured digital assets, consequently enhancing production efficiency and the intelligence level of decision-making.

7. Open Challenges and Future Research Directions

Despite the rapid progress of graph representation learning and its demonstrated effectiveness across a wide range of applications, several fundamental challenges remain unresolved. Addressing these issues is crucial for advancing the theoretical foundations, improving practical robustness, and broadening the applicability of graph learning models. In this section, we outline several promising and open research directions that warrant further investigation.

7.1. Theoretical Understanding of Representation Expressivity

Although numerous graph representation learning models have been proposed, a rigorous theoretical understanding of their expressive power remains limited. Existing studies have shown that many message-passing graph neural networks are constrained by the Weisfeiler-Lehman test and struggle to distinguish certain non-isomorphic graph structures [245,246]. However, expressivity is not solely determined by theoretical distinguishability, but also by how structural and attribute information is encoded, propagated, and aggregated in practice.

From a methodological perspective, future research should aim to establish more refined theoretical frameworks that characterize the relationship between model architecture, receptive field design, and task-specific expressivity. In particular, understanding how node-level, subgraph-level, and global structural patterns are captured under different aggregation mechanisms, as well as how architectural choices interact with optimization objectives, remains an open problem [247,248]. Such insights would not only clarify the limitations of existing models, but also guide the principled design of more expressive and interpretable graph learning architectures.

7.2. Generalization across Graphs and Domains

Most existing graph representation learning methods are evaluated under transductive or single-graph settings, where training and testing are performed on the same graph or closely related graphs. However, real-world applications often require models to generalize across different graphs, domains, or distributions, such as transferring knowledge between networks of varying scales, structures, or semantic contexts [249,250].

Achieving robust cross-graph and cross-domain generalization poses significant challenges due to structural heterogeneity, distribution shifts, and domain-specific biases. At the methodological level, future research may explore domain-invariant representations, graph-level normalization strategies, and meta-learning or continual learning paradigms tailored for graphs [251,252]. In addition, establishing standardized benchmarks and evaluation protocols for inductive and cross-domain graph learning would be instrumental in systematically assessing generalization performance.

7.3. Learning under Limited Supervision and Noisy Data

Obtaining high-quality annotations for graph-structured data is often expensive, incomplete, or unreliable, particularly in large-scale or dynamic networks. While recent advances in self-supervised and contrastive learning have alleviated the reliance on labeled data, learning robust representations under limited supervision and noisy conditions remains a critical challenge [253,254].

From a technical perspective, future research may focus on developing robustness-oriented learning paradigms that explicitly account for label noise, partial supervision, and distributional uncertainty in graph data. Promising directions include noise-aware objective functions that distinguish reliable from unreliable supervision, consistency-based regularization strategies that enforce representation stability under perturbations or data augmentation, and adaptive training mechanisms that dynamically regulate the influence of noisy samples during optimization [255–257]. In addition, incorporating uncertainty estimation, weak supervision signals, and prior domain knowledge into graph learning frameworks offers a principled way to improve model reliability and generalization in realistic, imperfect data environments.

7.4. Integration with Foundation Models and Multimodal Learning

The emergence of large-scale foundation models has transformed representation learning in language, vision, and multimodal domains, yet their integration with graph-structured data is still in its early stages [258,259]. Graphs provide a natural abstraction for modeling relational inductive biases, while foundation models offer powerful semantic understanding and generalization capabilities [260].

In terms of methodological integration, a key future direction lies in developing unified frameworks that seamlessly integrate graph representation learning with foundation models, enabling joint reasoning over structural, textual, visual, and temporal modalities [261]. Challenges include aligning heterogeneous representations, scaling graph computations to large contexts, and designing efficient interfaces between graph encoders and pretrained models [262]. Progress in this direction could significantly enhance the expressiveness and applicability of graph learning systems in complex, multimodal environments.

7.5. Trustworthy, Fair, and Ethical Graph Learning

As graph representation learning is increasingly deployed in high-stakes applications such as social networks, recommender systems, and biological analysis, concerns regarding trustworthiness, fairness, and ethical implications have become increasingly prominent. Graph models may inadvertently amplify biases present in data, exhibit vulnerability to adversarial perturbations, or lack transparency in their decision-making processes [263,264].

At the methodological and system-design level, future research should focus on developing explainable graph learning methods, fairness-aware objectives, and robustness guarantees against adversarial or spurious structural patterns [265,266]. In addition, establishing ethical guidelines and evaluation criteria for responsible graph learning, particularly in sensitive application domains, remains an important and largely unexplored area. Addressing these challenges is essential for ensuring that graph representation learning methods are not only effective, but also reliable and socially responsible.

Author Contributions

L.H.: conceptualization, methodology; Y.Y., D.L., Z.C., and Q.S.: writing—original draft preparation; M.Z., H.Y., and Y.C.: visualization, investigation; L.H.: supervision, writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported in part by the National Natural Science Foundation of China under grants 62373348, in part by the Tianshan Talent Training Program under grant 2023TSYCLJ0021, and in part by the Pioneer Hundred Talents Program of Chinese Academy of Sciences.

Data Availability Statement

Not applicable.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments.

Conflicts of Interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the author(s) used ChatGPT to assist with rephrasing and making grammatical corrections. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

References

1. D'Souza, R.M.; Di Bernardo, M.; Liu, Y.Y. Controlling Complex Networks with Complex Nodes. *Nat. Rev. Phys.* **2023**, *5*, 250–262.
2. Rácz, M.Z.; Rigobon, D.E. Towards Consensus: Reducing Polarization by Perturbing Social Networks. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 3450–3464.
3. Teji, B.; Roy, S.; Guzzi, P.H.; et al. Application of Generative Graph Models in Biological Network Regeneration: A Selective Review and Qualitative Analysis. In Proceedings of the 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Lisbon, Portugal, 3–6 December 2024; pp. 5786–5792.
4. Du, Y.L.; Yi, T.H.; Li, X.J.; et al. Advances in Intellectualization of Transportation Infrastructures. *Engineering* **2023**, *24*, 239–252.
5. Alves, J.; Sousa, P.; Cruz, T.; et al. A Review of Architecture Features for Distributed and Resilient Industrial Cyber–Physical Systems. *J. Manuf. Syst.* **2025**, *82*, 1069–1090.
6. Artime, O.; Grassia, M.; De Domenico, M.; et al. Robustness and Resilience of Complex Networks. *Nat. Rev. Phys.* **2024**, *6*, 114–131.
7. Ji, P.; Ye, J.; Mu, Y.; et al. Signal Propagation in Complex Networks. *Phys. Rep.* **2023**, *1017*, 1–96.
8. Pei, X.; Deng, X.; Xiong, N.N.; et al. Complex Graph Analysis and Representation Learning: Problems, Techniques, and Applications. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 4990–5007.
9. Ding, J.; Liu, C.; Zheng, Y.; et al. A Comprehensive Survey on Artificial Intelligence for Complex Network: Potential, Methodology and Application. *arXiv* **2024**, arXiv:2402.16887.
10. Wang, K.; An, J.; Zhou, M.; et al. Minority-Weighted Graph Neural Network for Imbalanced Node Classification in Social Networks of Internet of People. *IEEE Internet Things J.* **2023**, *10*, 330–340.
11. Khoshraftar, S.; An, A. A Survey on Graph Representation Learning Methods. *ACM Trans. Intell. Syst. Technol.* **2024**, *15*, 1–55.
12. Tan, X.; Zhou, Y.; Zhou, M.; et al. Learning to Detect Critical Nodes in Sparse Graphs via Feature Importance Awareness. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 3772–3782.
13. Shi, B.S.; Wang, Y.Q.; Guo, F.D.; et al. Domain Adaptation for Graph Representation Learning: Challenges, Progress, and Prospects. *J. Comput. Sci. Technol.* **2025**, 40, 283–300.
14. Liao, J.; Xiong, P.; Zhou, M.; et al. Adversarial Subgraph Contrastive Learning for Predicting Grasp Stability of Robotic Hands With Multimodal Signals. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 17720–17733.
15. Bozorgi, E.; Alqaiidi, S.K.; Shams, A.; et al. A Survey on Recent Random Walk-based Methods for Embedding Knowledge Graphs. *arXiv* **2024**, arXiv:2406.07402.
16. Hamedani, M.R.; Ryu, J.S.; Kim, S.W. Random-Walk-Based or Similarity-Based Methods, Which is Better

for Directed Graph Embedding? In Proceedings of the 2024 IEEE International Conference on Big Data and Smart Computing (BigComp), Bangkok, Thailand, 18–21 February 2024; pp. 83–89.

17. Saberi-Movahed, F.; Berahmand, K.; Sheikhpour, R.; et al. Nonnegative Matrix Factorization in Dimensionality Reduction: A Survey. *ACM Comput. Surv.* **2025**, *58*, 1–41.
18. Waikhom, L.; Patgiri, R. A Survey of Graph Neural Networks in Various Learning Paradigms: Methods, Applications, and Challenges. *Artif. Intell. Rev.* **2023**, *56*, 6295–6364.
19. Sun, C.; Li, C.; Lin, X.; et al. Attention-based Graph Neural Networks: A Survey. *Artif. Intell. Rev.* **2023**, *56*, 2263–2310.
20. Zhao, Z.; Yang, Z.; Li, C.; et al. Dual Feature Interaction-Based Graph Convolutional Network. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 9019–9030.
21. Malla, A.M.; Banka, A.A. Comprehensive Review of Graph Neural Networks: Challenges, Classification, Architectures, Applications, and Potential Utility in Bioinformatics. *Expert Syst.* **2025**, *42*, e70091.
22. Zhao, Z.; Zhou, H.; Qi, L.; et al. Inductive Representation Learning via CNN for Partially-Unseen Attributed Networks. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 695–706.
23. Berahmand, K. Deep Graph Representation Learning for Complex Networks Applications. PhD thesis, Queensland University of Technology, Brisbane, Australia, 2025.
24. Chen, L.; Li, Y.; Liu, H.; et al. Node and Edge Joint Embedding for Heterogeneous Information Network. *Big Data Min. Anal.* **2024**, *7*, 730–752.
25. Mijatovic, G.; Antonacci, Y.; Javorka, M.; et al. Network Representation of Higher-Order Interactions Based on Information Dynamics. *IEEE Trans. Netw. Sci. Eng.* **2025**, *12*, 1872–1884.
26. Cai, L.; Mao, X.; Zhou, Y.; et al. A Survey on Temporal Knowledge Graph: Representation Learning and Applications. *arXiv* **2024**, arXiv:2403.04782.
27. Xia, R.; Liu, H.; Li, A.; et al. Incomplete Graph Learning: A Comprehensive Survey. *Neural Netw.* **2025**, p. 107682.
28. Bing, R.; Yuan, G.; Zhu, M.; et al. Heterogeneous Graph Neural Networks Analysis: A Survey of Techniques, Evaluations and Applications. *Artif. Intell. Rev.* **2023**, *56*, 8003–8042.
29. Li, Z.; Yu, J.; Zhang, G.; et al. Dynamic Spatio-Temporal Graph Network with Adaptive Propagation Mechanism for Multivariate Time Series Forecasting. *Expert Syst. Appl.* **2023**, *216*, 119374.
30. Antelmi, A.; Cordasco, G.; Polato, M.; et al. A Survey on Hypergraph Representation Learning. *ACM Comput. Surv.* **2023**, *56*, 1–38.
31. Shou, Y.; Cao, X.; Meng, D. Spegcl: Self-Supervised Graph Spectrum Contrastive Learning without Positive Samples. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**.
32. Luo, Z.; Dong, Y.; Zheng, Q.; et al. Dual-Channel Graph Contrastive Learning for Self-Supervised Graph-Level Representation Learning. *Pattern Recognit.* **2023**, *139*, 109448.
33. Girvan, M.; Newman, M.E. Community Structure in Social and Biological Networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826.
34. Lancichinetti, A.; Fortunato, S.; Radicchi, F.; et al. Benchmark Graphs for Testing Community Detection Algorithms. *Phys. Rev. E* **2008**, *78*, 046110.
35. Sen, P.; Namata, G.; Bilgic, M.; et al. Collective Classification in Network Data. *AI Mag.* **2008**, *29*, 93–93.
36. Yang, J.; Leskovec, J. Defining and Evaluating Network Communities Based on Ground-Truth. In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, New York, NY, USA, 12–16 August 2012; pp. 1–8.
37. Wang, X.; Ji, H.; Shi, C.; et al. Heterogeneous Graph Attention Network. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2022–2032.
38. Yang, C.; Liu, Z.; Zhao, D.; et al. Network Representation Learning with Rich Text Information. In Proceedings of the IJCAI, Buenos Aires, Argentina, 25–31 July 2015; Volume 2015, pp. 2111–2117.
39. Adamic, L.A.; Glance, N. The Political Blogosphere and the 2004 US Election: Divided They Blog. In Proceedings of the 3rd International Workshop on Link Discovery, Chicago, IL, USA, 21–25 August 2005; pp. 36–43.
40. Yang, Y.; Hu, L.; Li, G.; et al. Link-Based Attributed Graph Clustering via Approximate Generative Bayesian Learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2025**, *55*, 5730–5743.
41. Huang, X.; Li, J.; Hu, X. Accelerated Attributed Network Embedding. In Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, TX, USA, 27–29 April 2017; pp. 633–641.

42. Wale, N.; Watson, I.A.; Karypis, G. Comparison of Descriptor Spaces for Chemical Compound Retrieval and Classification. *Knowl. Inf. Syst.* **2008**, *14*, 347–375.
43. Debnath, A.K.; Lopez de Compadre, R.L.; Debnath, G.; et al. Structure-Activity Relationship of Mutagenic Aromatic and Heteroaromatic Nitro Compounds. Correlation with Molecular Orbital Energies and Hydrophobicity. *J. Med. Chem.* **1991**, *34*, 786–797.
44. Dobson, P.D.; Doig, A.J. Distinguishing Enzyme Structures from Non-Enzymes without Alignments. *J. Mol. Biol.* **2003**, *330*, 771–783.
45. Toivonen, H.; Srinivasan, A.; King, R.D.; et al. Statistical Evaluation of the Predictive Toxicology Challenge 2000–2001. *Bioinformatics* **2003**, *19*, 1183–1193.
46. Ramakrishnan, R.; Dral, P.O.; Rupp, M.; et al. Quantum Chemistry Structures and Properties of 134 Kilo Molecules. *Sci. Data* **2014**, *1*, 1–7.
47. Chen, G.; Chen, P.; Hsieh, C.Y.; et al. Alchemy: A Quantum Chemistry Dataset for Benchmarking AI Models. *arXiv* **2019**, arXiv:1906.09427.
48. Zitnik, M.; Leskovec, J. Predicting Multicellular Function Through Multi-Layer Tissue Networks. *Bioinformatics* **2017**, *33*, i190–i198.
49. Newman, M.E. Modularity and Community Structure in Networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582.
50. Liu, Y.; Yang, X.; Zhou, S.; et al. Simple Contrastive Graph Clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 13789–13800.
51. Gori, M.; Monfardini, G.; Scarselli, F. A New Model for Learning in Graph Domains. Proceedings. 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005, Volume 2, pp. 729–734.
52. Schaeffer, S.E. Graph Clustering. *Comput. Sci. Rev.* **2007**, *1*, 27–64.
53. Fabbri, R.; Costa, L.D.F.; Torelli, J.C.; et al. 2D Euclidean Distance Transform Algorithms: A Comparative Survey. *ACM Comput. Surv.* **2008**, *40*, 1–44.
54. Mienye, I.D.; Swart, T.G.; Obaido, G. Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information* **2024**, *15*, 517.
55. Purwono, P.; Ma'arif, A.; Rahmaniar, W.; et al. Understanding of Convolutional Neural Network (CNN): A Review. *Int. J. Robot. Control Syst.* **2022**, *2*, 739–748.
56. Li, X.; Sun, L.; Ling, M.; Peng, Y. A Survey of Graph Neural Network Based Recommendation in Social Networks. *Neurocomputing* **2023**, *549*, 126441.
57. Li, L.; Zhang, Y.; Wang, G.; et al. Kolmogorov–Arnold Graph Neural Networks for Molecular Property Prediction. *Nat. Mach. Intell.* **2025**, *7*, 1346–1354.
58. Bai, X.; He, S.; Li, Y.; et al. Construction of a Knowledge Graph for Framework Material Enabled by Large Language Models and Its Application. *npj Comput. Mater.* **2025**, *11*, 51.
59. Tang, M.; Li, B.; Chen, H. Application of Message Passing Neural Networks for Molecular Property Prediction. *Curr. Opin. Struct. Biol.* **2023**, *81*, 102616.
60. Li, Y.; Qiao, G.; Gao, X.; et al. Supervised Graph Co-Contrastive Learning for Drug–Target Interaction Prediction. *Bioinformatics* **2022**, *38*, 2847–2854.
61. Veličković, P. Everything is Connected: Graph Neural Networks. *Curr. Opin. Struct. Biol.* **2023**, *79*, 102538.
62. Gao, C.; Zheng, Y.; Li, N.; et al. A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *ACM Trans. Recomm. Syst.* **2023**, *1*, 1–51.
63. Li, Y.; Yu, D.; Liu, Z.; et al. Graph Neural Network for Spatiotemporal Data: Methods and Applications. *arXiv* **2023**, arXiv:2306.00012.
64. Bhatti, U.A.; Tang, H.; Wu, G.; et al. Deep Learning with Graph Convolutional Networks: An Overview and Latest Applications in Computational Intelligence. *Int. J. Intell. Syst.* **2023**, *2023*, 8342104.
65. Oh, D.; Moon, J.; Park, K.; et al. GCN-Assisted Attention-Guided UNet for Automated Retinal OCT Segmentation. *Expert Syst. Appl.* **2024**, *249*, 123620.
66. Dong, Y.; Liu, Q.; Du, B.; et al. Weighted Feature Fusion of Convolutional Neural Network and Graph Attention Network for Hyperspectral Image Classification. *IEEE Trans. Image Process.* **2022**, *31*, 1559–1572.
67. Velickovic, P.; Cucurull, G.; Casanova, A.; et al. Graph Attention Networks. *Stat* **2017**, *1050*, 10–48550.
68. Zhai, H.; Hou, H.; Luo, J.; et al. DGDTA: Dynamic Graph Attention Network for Predicting Drug–Target Binding Affinity. *BMC Bioinform.* **2023**, *24*, 367.

69. Lv, Q.; Chen, G.; Yang, Z.; et al. Meta Learning with Graph Attention Networks for Low-Data Drug Discovery. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 11218–11230.
70. Meng, L.; Zhang, H. GACT-PPIS: Prediction of Protein-Protein Interaction Sites Based on Graph Structure and Transformer Network. *Int. J. Biol. Macromol.* **2024**, *283*, 137272.
71. Tanvir, R.B.; Islam, M.M.; Sobhan, M.; et al. Mogat: A Multi-Omics Integration Framework Using Graph Attention Networks for Cancer Subtype Prediction. *Int. J. Mol. Sci.* **2024**, *25*, 2788.
72. Leskovec, J.; Faloutsos, C. Sampling from Large Graphs. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; pp. 631–636.
73. Zeng, H.; Zhou, H.; Srivastava, A.; Kannan, R.; Prasanna, V. GraphSaint: Graph Sampling Based Inductive Learning Method. *arXiv* **2019**, arXiv:1907.04931.
74. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive Representation Learning on Large Graphs. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
75. Corso, G.; Stark, H.; Jegelka, S.; et al. Graph Neural Networks. *Nat. Rev. Methods Primers* **2024**, *4*, 17–43.
76. Liu, J.; Yang, C.; Lu, Z.; et al. Graph Foundation Models: Concepts, Opportunities and Challenges. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 5023–5044.
77. Kipf, T. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
78. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online Learning of Social Representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
79. Grover, A.; Leskovec, J. node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
80. Hamilton, W.L. *Graph Representation Learning*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2020.
81. Zhang, Z.; Cui, P.; Zhu, W. Deep Learning on Graphs: A Survey. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 249–270.
82. Zhao, B.W.; Su, X.R.; Hu, P.W.; et al. iGRLDTI: An Improved Graph Representation Learning Method for Predicting Drug–Target Interactions over Heterogeneous Biological Information Network. *Bioinformatics* **2023**, *39*, btad451.
83. He, Y.; Yang, Y.; Su, X.; et al. Incorporating Higher Order Network Structures to Improve miRNA–Disease Association Prediction Based on Functional Modularity. *Brief. Bioinform.* **2023**, *24*, bbac562.
84. Zhang, M.L.; Zhao, B.W.; Su, X.R.; et al. RLFDDA: A Meta-Path Based Graph Representation Learning Model for Drug–Disease Association Prediction. *BMC Bioinform.* **2022**, *23*, 516.
85. Sun, Y.; Han, J.; Yan, X.; et al. Pathsim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *Proc. VLDB Endow.* **2011**, *4*, 992–1003.
86. Shi, C.; Kong, X.; Huang, Y.; et al. Hetesim: A General Framework for Relevance Measure in Heterogeneous Networks. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 2479–2492.
87. Dong, Y.; Chawla, N.V.; Swami, A. Metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 135–144.
88. Fu, X.; Zhang, J.; Meng, Z.; et al. Magnn: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 2331–2341.
89. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; et al. Modeling Relational Data with Graph Convolutional Networks. In Proceedings of the European Semantic Web Conference, Heraklion, Greece, 3–7 June 2018; pp. 593–607.
90. Vashishth, S.; Sanyal, S.; Nitin, V.; et al. Composition-Based Multi-Relational Graph Convolutional Networks. *arXiv* **2019**, arXiv:1911.03082.
91. Hu, Z.; Dong, Y.; Wang, K.; et al. Heterogeneous Graph Transformer. In Proceedings of the Web Conference 2020,Taipei, Taiwan, 20–24 April 2020; pp. 2704–2710.
92. Milo, R.; Shen-Orr, S.; Itzkovitz, S.; et al. Network Motifs: Simple Building Blocks of Complex Networks. *Science* **2002**, *298*, 824–827.
93. Alon, U. Network Motifs: Theory and Experimental Approaches. *Nat. Rev. Genet.* **2007**, *8*, 450–461.

94. Henderson, K.; Gallagher, B.; Eliassi-Rad, T.; et al. Rolx: Structural Role Extraction & Mining in Large Graphs. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 1231–1239.
95. Monti, F.; Otness, K.; Bronstein, M.M. Motifnet: A Motif-Based Graph Convolutional Network for Directed Graphs. In Proceedings of the 2018 IEEE Data Science Workshop (DSW), Lausanne, Switzerland, 4–6 June 2018; pp. 225–228.
96. Chen, X.; Cai, R.; Fang, Y.; et al. Motif Graph Neural Network. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 14833–14847.
97. Pržulj, N. Biological Network Comparison Using Graphlet Degree Distribution. *Bioinformatics* **2007**, *23*, e177–e183.
98. Shervashidze, N.; Vishwanathan, S.V.N.; Petri, T.H.; et al. Efficient Graphlet Kernels for Large Graph Comparison. In Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, Clearwater Beach, FL, USA, 16–18 April 2009; Volume 5, pp. 488–495.
99. Sun, L.; Huang, Z.; Wang, Z.; et al. Motif-Aware Riemannian Graph Neural Network with Generative-Contrastive Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 26-27 February 2024, Volume 38; pp. 9044–9052.
100. Inae, E.; Liu, G.; Jiang, M. Motif-Aware Attribute Masking for Molecular Graph Pre-Training. *arXiv* **2023**, arXiv:2309.04589.
101. Zhang, S.; Hu, Z.; Subramonian, A.; et al. Motif-Driven Contrastive Learning of Graph Representations. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 4063–4075.
102. Ribeiro, L.F.; Saverese, P.H.; Figueiredo, D.R. Struc2vec: Learning Node Representations from Structural Identity. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 385–394.
103. Donnat, C.; Zitnik, M.; Hallac, D.; et al. Learning Structural Node Embeddings via Diffusion Wavelets. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1320–1329.
104. Ahmed, N.K.; Rossi, R.; Lee, J.B.; et al. Learning Role-Based Graph Embeddings. *arXiv* **2018**, arXiv:1802.02896.
105. Heidrich, M.; Heidemann, J.; Buchkremer, R.; et al. Ffstruc2vec: Flat, Flexible and Scalable Learning of Node Representations from Structural Identities. *arXiv* **2025**, arXiv:2504.01122.
106. Zhu, W.; Wen, T.; Song, G.; Wang, L.; Zheng, B.; et al. On Structural Expressive Power of Graph Transformers. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 3628–3637.
107. Cantürk, S.; Liu, R.; Lapointe-Gagné, O.; et al. Graph Positional and Structural Encoder. *arXiv* **2023**, arXiv:2307.07107.
108. Hu, L.; Zhang, M.; Hu, P.; et al. Dual-Channel Hypergraph Convolutional Network for Predicting Herb–Disease Associations. *Brief. Bioinform.* **2024**, *25*, bbae067.
109. Feng, Y.; You, H.; Zhang, Z.; et al. Hypergraph Neural Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 29–31 January 2019; Volume 33; pp. 3558–3565.
110. Yadati, N.; Nimishakavi, M.; Yadav, P.; et al. Hypergcn: A New Method for Training Graph Convolutional Networks on Hypergraphs. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
111. Huang, J.; Yang, J. Unignn: A Unified Framework for Graph and Hypergraph Neural Networks. *arXiv* **2021**, arXiv:2105.00956.
112. Chien, E.; Pan, C.; Peng, J.; et al. You are Allset: A Multiset Function Framework for Hypergraph Neural Networks. *arXiv* **2021**, arXiv:2106.13264.
113. Saifuddin, K.M.; Aktas, M.E.; et al. Topology-Guided Hypergraph Transformer Network: Unveiling Structural Insights for Improved Representation. *arXiv* **2023**, arXiv:2310.09657.
114. Kim, S.; Lee, S.Y.; Gao, Y.; et al. A Tutorial on Hypergraph Neural Networks: An In-Depth and Step-by-Step Guide. In Proceedings of the 34th ACM International Conference on Information and Knowledge Management, Seoul, Korea, 10–14 November 2025; pp. 6829–6832.
115. Qu, S.; Wang, W.; Li, Y.F.; et al. Hypergraph Node Representation Learning with One-Stage Message Passing. *Neural Netw.* **2025**, p. 107973.
116. Ebli, S.; Defferrard, M.; Spreemann, G. Simplicial Neural Networks. *arXiv* **2020**, arXiv:2010.03633.

117. Wu, H.; Yip, A.; Long, J.; et al. Simplicial Complex Neural Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *46*, 561–575.
118. Battiloro, C.; Testa, L.; Giusti, L.; et al. Generalized Simplicial Attention Neural Networks. *IEEE Trans. Signal Inf. Process. Netw.* **2024**.
119. Hajij, M.; Papillon, M.; Frantzen, F.; et al. TopoX: A Suite of Python Packages for Machine Learning on Topological Domains. *J. Mach. Learn. Res.* **2024**, *25*, 1–8.
120. Pareja, A.; Domeniconi, G.; Chen, J.; et al. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. In Proceedings of the 2020 AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34; pp. 5363–5370.
121. Sankar, A.; Wu, Y.; Gou, L.; et al. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 519–527.
122. Karmim, Y.; Lafon, M.; Fournier-S'Niehotta, R.; et al. Supra-Laplacian Encoding for Transformer on Dynamic Graphs. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 17215–17246.
123. Yu, L.; Sun, L.; Du, B.; et al. Towards Better Dynamic Graph Learning: New Architecture and Unified Library. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 67686–67700.
124. Cong, W.; Zhang, S.; Kang, J.; et al. Do We Really Need Complicated Model Architectures for Temporal Networks? *arXiv* **2023**, arXiv:2302.11636.
125. Wu, Y.; Fang, Y.; Liao, L. On the Feasibility of Simple Transformer for Dynamic Graph Modeling. In Proceedings of the ACM Web Conference 2024, Singapore, 13–17 May 2024; pp. 870–880.
126. Li, J.; Wu, R.; Jin, X.; et al. State Space Models on Temporal Graphs: A First-Principles Study. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 127030–127058.
127. Chen, H.; Jiao, P.; Du, M.; et al. TGFormer: Towards Temporal Graph Transformer with Auto-Correlation Mechanism. *Pattern Recognit.* **2026**, *170*, 112053.
128. Poursafaei, F.; Huang, S.; Pelrine, K.; et al. Towards Better Evaluation for Dynamic Link Prediction. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 32928–32941.
129. Huang, S.; Poursafaei, F.; Danovitch, J.; et al. Temporal Graph Benchmark for Machine Learning on Temporal Graphs. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 2056–2073.
130. Gastinger, J.; Huang, S.; Galkin, M.; et al. TGB 2.0: A Benchmark for Learning on Temporal Knowledge Graphs and Heterogeneous Graphs. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 140199–140229.
131. Jiao, P.; Chen, H.; Guo, X.; et al. A Survey on Temporal Interaction Graph Representation Learning: Progress, Challenges, and Opportunities. *arXiv* **2025**, arXiv:2505.04461.
132. Xu, D.; Ruan, C.; Korpeoglu, E.; et al. Inductive Representation Learning on Temporal Graphs. *arXiv* **2020**, arXiv:2002.07962.
133. Wang, Z.; Zhou, S.; Chen, J.; et al. Dynamic Graph Transformer with Correlated Spatial-Temporal Positional Encoding. In Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, Hannover, Germany, 10–14 March 2025; pp. 60–69.
134. Tieu, K.; Fu, D.; Li, Z.; et al. Learnable Spatial-Temporal Positional Encoding for Link Prediction. *arXiv* **2025**, arXiv:2506.08309.
135. Li, Y.; Wu, Y.; Sun, M.; et al. Learning Continuous Dynamic Network Representation with Transformer-Based Temporal Graph Neural Network. *Inf. Sci.* **2023**, *649*, 119596.
136. Rossi, E.; Chamberlain, B.; Frasca, F.; et al. Temporal Graph Networks for Deep Learning on Dynamic Graphs. *arXiv* **2020**, arXiv:2006.10637.
137. Kumar, S.; Zhang, X.; Leskovec, J. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1269–1278.
138. Wang, Y.; Chang, Y.Y.; Liu, Y.; et al. Inductive Representation Learning in Temporal Networks via Causal Anonymous Walks. *arXiv* **2021**, arXiv:2101.05974.
139. Su, J.; Zou, D.; Wu, C. PRES: Toward Scalable Memory-Based Dynamic Graph Neural Networks. *arXiv* **2024**, arXiv:2402.04284.
140. Longa, A.; Lachi, V.; Santin, G.; et al. Graph Neural Networks for Temporal Graphs: State of the Art, Open Challenges, and Opportunities. *arXiv* **2023**, arXiv:2302.01018.
141. Zheng, Y.; Yi, L.; Wei, Z. A Survey of Dynamic Graph Neural Networks. *Front. Comput. Sci.* **2025**, *19*, 196323.

142. Cappart, Q.; Chételat, D.; Khalil, E.B.; et al. Combinatorial Optimization and Reasoning with Graph Neural Networks. *J. Mach. Learn. Res.* **2023**, *24*, 1–61.
143. Ammar, K.; Inoubli, W.; Zghal, S.; et al. Navigating Complexity: A Comprehensive Review of Heterogeneous Information Networks and Embedding Techniques. *Knowl. Inf. Syst.* **2025**, *67*, 4703–4736.
144. Song, Z.; Yang, X.; Xu, Z.; et al. Graph-Based Semi-Supervised Learning: A Comprehensive Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 8174–8194.
145. Das, S.S.; Arafat, N.A.; Rahman, M.; et al. SGS-GNN: A Supervised Graph Sparsification Method for Graph Neural Networks. *arXiv* **2025**, arXiv:2502.10208.
146. Lv, X.; Leng, C.; Peng, J.; et al. Incremental Semi-Supervised Graph Learning NMF with Block-Diagonal. *Eng. Appl. Artif. Intell.* **2024**, *130*, 107668.
147. Ai, W.; Li, J.; Wang, Z.; et al. Contrastive Multi-Graph Learning with Neighbor Hierarchical Sifting for Semi-Supervised Text Classification. *Expert Syst. Appl.* **2025**, *266*, 125952.
148. Xie, Y.; Xu, Z.; Zhang, J.; et al. Self-Supervised Learning of Graph Neural Networks: A Unified Review. *arXiv* **2022**, arXiv:2102.10757.
149. Ju, W.; Wang, Y.; Qin, Y.; et al. Towards Graph Contrastive Learning: A Survey and Beyond. *arXiv* **2024**, arXiv:2405.11868.
150. Liu, Y.; Jin, M.; Pan, S.; et al. Graph Self-Supervised Learning: A Survey. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 5879–5900.
151. Sattarzadeh, A.R.; Pathirana, P.N.; Palaniswami, M. Implicit Sensing Self-Supervised Learning Based on Graph Multi-Pretext Tasks for Traffic Flow Prediction. *Neural Comput. Appl.* **2025**, *37*, 1041–1066.
152. Zhao, Z.; Su, Y.; Li, Y.; et al. A Survey on Self-Supervised Graph Foundation Models: Knowledge-Based Perspective. *IEEE Trans. Knowl. Data Eng.* **2025**, *37*, 4389–4410.
153. Wang, S.; Yang, R.; Lai, Y.; et al. Rethinking Semi-Supervised Node Classification with Self-Supervised Graph Clustering. *arXiv* **2025**, arXiv:2511.19976.
154. Liu, N.; Wang, X.; Bo, D.; et al. Revisiting Graph Contrastive Learning from the Perspective of Graph Spectrum. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 2972–2983.
155. Shuai, J.; Zhang, K.; Wu, L.; et al. A Review-Aware Graph Contrastive Learning Framework for Recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1283–1293.
156. Yang, Y.; Wu, Z.; Wu, L.; et al. Generative-Contrastive Graph Learning for Recommendation. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023; pp. 1117–1126.
157. Guo, X.; Wang, Y.; Wei, Z.; et al. Architecture Matters: Uncovering Implicit Mechanisms in Graph Contrastive Learning. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 28585–28610.
158. Ju, W.; Fang, Z.; Gu, Y.; et al. A Comprehensive Survey on Deep Graph Representation Learning. *Neural Netw.* **2024**, *173*, 106207.
159. Xia, F.; Peng, C.; Ren, J.; et al. Graph Learning. *Found. Trends Signal Process.* **2025**, *19*, 371–551.
160. Guo, X.; Zhao, L. A Systematic Survey on Deep Generative Models for Graph Generation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 5370–5390.
161. Li, J.; Wu, R.; Zhu, Y.; et al. Revisiting and Benchmarking Graph Autoencoders: A Contrastive Learning Perspective. *arXiv* **2024**, arXiv:2410.10241.
162. Han, Y.; Hui, L.; Jiang, H.; et al. Generative Subgraph Contrast for Self-Supervised Graph Representation Learning. In Proceedings of the 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; pp. 91–107.
163. Zhou, J.; Liu, L.; Wei, W.; et al. Network Representation Learning: From Preprocessing, Feature Extraction to Node Embedding. *ACM Comput. Surv.* **2022**, *55*, 1–35.
164. Hou, Z.; Liu, X.; Cen, Y.; et al. GraphMAE: Self-Supervised Masked Graph Autoencoders. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 594–604.
165. Tian, Y.; Zhang, C.; Kou, Z.; et al. UGMAE: A Unified Framework for Graph Masked Autoencoders. *arXiv* **2024**, arXiv:2402.08023.
166. Zhao, Z.; Li, Y.; Zou, Y.; et al. Masked Graph Autoencoder with Non-Discrete Bandwidths. In Proceedings of the ACM Web Conference 2024, Singapore, 13–17 May 2024; pp. 377–388.

167. Tan, Q.; Liu, N.; Huang, X.; et al. S2GAE: Self-Supervised Graph Autoencoders Are Generalizable Learners with Graph Masking. In Proceedings of the 16th ACM International Conference on Web Search and Data Mining, Singapore, 27 February–3 March 2023; pp. 787–795.
168. Jiang, W.; Liu, H.; Xiong, H. When Graph Neural Network Meets Causality: Opportunities, Methodologies and an Outlook. *arXiv* **2024**, arXiv:2312.12477.
169. Xu, Z.; Ma, K.; Liu, Y.; et al. Causal Representation Learning for Robust Anomaly Detection in Complex Environments. In Proceedings of the 2025 2nd International Conference on Digital Economy, Blockchain and Artificial Intelligence, Beijing, China, 27–29 June 2025; pp. 382–387.
170. Gao, H.; Li, J.; Qiang, W.; et al. Robust Causal Graph Representation Learning against Confounding Effects. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023, Volume 37; pp. 7624–7632.
171. Wang, W.; Lin, X.; Feng, F.; et al. Causal Representation Learning for Out-of-Distribution Recommendation. In Proceedings of the ACM Web Conference 2022, Austin, TX, USA, 30 April–4 May 2022; pp. 3562–3571.
172. Ahuja, K.; Mahajan, D.; Wang, Y.; et al. Interventional Causal Representation Learning. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 372–407.
173. Jaimini, U.; Sheth, A. CausalKG: Causal Knowledge Graph Explainability Using Interventional and Counterfactual Reasoning. *IEEE Internet Comput.* **2022**, *26*, 43–50.
174. Job, S.; Tao, X.; Cai, T.; et al. Exploring Causal Learning Through Graph Neural Networks: An In-Depth Review. *WIREs Data Min. Knowl. Discov.* **2025**, *15*, e70024.
175. Li, G.; Su, X.; Yang, Y.; et al. Interpretability-Guided RNA N6-Methyladenosine Modification Site Prediction with Invertible Neural Networks. *Commun. Biol.* **2025**, *8*, 1022.
176. Li, M.M.; Huang, K.; Zitnik, M. Graph Representation Learning in Biomedicine and Healthcare. *Nat. Biomed. Eng.* **2022**, *6*, 1353–1369.
177. Hu, L.; Zhang, J.; Pan, X.; et al. HiSCF: Leveraging Higher-Order Structures for Clustering Analysis in Biological Networks. *Bioinformatics* **2021**, *37*, 542–550.
178. Gaudelet, T.; Day, B.; Jamasb, A.R.; et al. Utilizing Graph Machine Learning within Drug Discovery and Development. *Brief. Bioinform.* **2021**, *22*, bbab159.
179. Hu, L.; Wang, X.; Huang, Y.A.; et al. A Survey on Computational Models for Predicting Protein-Protein Interactions. *Brief. Bioinform.* **2021**, *22*, bbab036.
180. Li, G.; Yang, Y.; Li, D.; et al. A Bijective Inference Network for Interpretable Identification of RNA N6-Methyladenosine Modification Sites. *Pattern Recognit.* **2025**, *164*, 111541.
181. Wu, S.; Sun, F.; Zhang, W.; et al. Graph Neural Networks in Recommender Systems: A Survey. *ACM Comput. Surv.* **2022**, *55*, 1–37.
182. Hu, L.; Pan, X.; Tang, Z.; et al. A Fast Fuzzy Clustering Algorithm for Complex Networks via a Generalized Momentum Method. *IEEE Trans. Fuzzy Syst.* **2021**, *30*, 3473–3485.
183. Yang, L.; Liu, Z.; Dou, Y.; et al. ConsisRec: Enhancing GNN for Social Recommendation via Consistent Neighbor Aggregation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Montreal, QC, Canada 11–15 July 2021; pp. 2141–2145.
184. Hu, L.; Yang, Y.; Tang, Z.; et al. FCAN-MOPSO: An Improved Fuzzy-Based Graph Clustering Algorithm for Complex Networks with Multiobjective Particle Swarm Optimization. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 3470–3484.
185. Jiang, W.; Luo, J. Graph Neural Network for Traffic Forecasting: A Survey. *Expert Syst. Appl.* **2022**, *207*, 117921.
186. Dolatyabi, P.; Khodayar, M. Graph Neural Networks and Their Applications in Power Systems: A Review. In Proceedings of the 2025 IEEE International Conference on Electro Information Technology (eIT), Valparaiso, Indiana, 29-31 May 2025; pp. 1–10.
187. Ji, S.; Pan, S.; Cambria, E.; et al. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514.
188. Zhou, J.; Cui, G.; Hu, S.; et al. Graph Neural Networks: A Review of Methods and Applications. *AI Open* **2020**, *1*, 57–81.
189. Su, X.; Hu, P.; Yi, H.; et al. Predicting Drug-Target Interactions over Heterogeneous Information Network. *IEEE J. Biomed. Health Inform.* **2022**, *27*, 562–572.
190. Zhao, B.W.; Su, X.R.; Yang, Y.; et al. Regulation-Aware Graph Learning for Drug Repositioning over Heterogeneous Biological Network. *Inf. Sci.* **2025**, *686*, 121360.

191. Zhao, B.W.; Wang, L.; Hu, P.W.; et al. Fusing Higher and Lower-Order Biological Information for Drug Repositioning via Graph Representation Learning. *IEEE Trans. Emerg. Top. Comput.* **2023**, *12*, 163–176.
192. Barabási, A.L.; Gulbahce, N.; Loscalzo, J. Network Medicine: A Network-Based Approach to Human Disease. *Nat. Rev. Genet.* **2011**, *12*, 56–68.
193. Reel, P.S.; Reel, S.; Pearson, E.; et al. Using Machine Learning Approaches for Multi-Omics Data Analysis: A Review. *Biotechnol. Adv.* **2021**, *49*, 107739.
194. Zitnik, M.; Nguyen, F.; Wang, B.; et al. Machine Learning for Integrating Data in Biology and Medicine: Principles, Practice, and Opportunities. *Inf. Fusion* **2019**, *50*, 71–91.
195. Yue, X.; Wang, Z.; Huang, J.; et al. Graph Embedding on Biomedical Networks: Methods, Applications and Evaluations. *Bioinformatics* **2020**, *36*, 1241–1251.
196. Yi, H.C.; You, Z.H.; Huang, D.S.; et al. Graph Representation Learning in Bioinformatics: Trends, Methods and Applications. *Brief. Bioinform.* **2022**, *23*, bbab340.
197. Zhang, X.M.; Liang, L.; Liu, L.; et al. Graph Neural Networks and Their Current Applications in Bioinformatics. *Front. Genet.* **2021**, *12*, 690049.
198. Li, G.; Zhao, B.; Su, X.; et al. Discovering Consensus Regions for Interpretable Identification of RNA N6-Methyladenosine Modification Sites via Graph Contrastive Clustering. *IEEE J. Biomed. Health Inform.* **2024**, *28*, 2362–2372.
199. Hu, L.; Hu, P.; Huang, Y.A. Special Issue on Graph Representation Learning in Drug Repositioning. *J. Comput. Biophys. Chem.* **2025**, *24*, 267–268.
200. Huang, K.; Chandak, P.; Wang, Q.; et al. A Foundation Model for Clinician-Centered Drug Repurposing. *Nat. Med.* **2024**, *30*, 3601–3613.
201. Ingraham, J.B.; Baranov, M.; Costello, Z.; et al. Illuminating Protein Space with a Programmable Generative Model. *Nature* **2023**, *623*, 1070–1078.
202. Cui, H.; Wang, C.; Maan, H.; et al. scGPT: Toward Building a Foundation Model for Single-Cell Multi-Omics Using Generative AI. *Nat. Methods* **2024**, *21*, 1470–1480.
203. Liu, S.; Nie, W.; Wang, C.; et al. Multi-Modal Molecule Structure-Text Model for Text-Based Retrieval and Editing. *Nat. Mach. Intell.* **2023**, *5*, 1447–1457.
204. Pan, X.; Hu, L.; Hu, P.; et al. Identifying Protein Complexes from Protein-Protein Interaction Networks Based on Fuzzy Clustering and GO Semantic Information. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *19*, 2882–2893.
205. Hu, L.; Zhang, J.; Pan, X.; et al. An Effective Link-Based Clustering Algorithm for Detecting Overlapping Protein Complexes in Protein-Protein Interaction Networks. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 3275–3289.
206. Wang, J.; Ma, A.; Chang, Y.; et al. scGNN is a Novel Graph Neural Network Framework for Single-Cell RNA-Seq Analyses. *Nat. Commun.* **2021**, *12*, 1882.
207. Su, X.; Hu, P.; Li, D.; et al. Interpretable Identification of Cancer Genes across Biological Networks via Transformer-Powered Graph Representation Learning. *Nat. Biomed. Eng.* **2025**, *9*, 371–389.
208. Singhal, V.; Chou, N.; Lee, J.; et al. BANKSY Unifies Cell Typing and Tissue Domain Segmentation for Scalable Spatial Omics Data Analysis. *Nat. Genet.* **2024**, *56*, 431–441.
209. Zhao, B.W.; He, Y.Z.; Su, X.R.; et al. Motif-Aware miRNA-Disease Association Prediction via Hierarchical Attention Network. *IEEE J. Biomed. Health Inform.* **2024**, *28*, 4281–4294.
210. Hu, L.; Chan, K.C. Discovering Variable-Length Patterns in Protein Sequences for Protein-Protein Interaction Prediction. *IEEE Trans. Nanobiosci.* **2015**, *14*, 409–416.
211. Su, X.; Hu, P.; You, Z.H.; et al. Dual-Channel Learning Framework for Drug-Drug Interaction Prediction via Relation-Aware Heterogeneous Graph Transformer. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 26–27 February 2024; Volume 38; pp. 249–256.
212. Zhao, B.W.; Su, X.R.; Li, D.X.; et al. A Graph Deep Learning-Based Framework for Drug-Disease Association Identification with Chemical Structure Similarities. *J. Comput. Biophys. Chem.* **2025**, *24*, 331–343.
213. Su, X.; Zhao, B.; Li, G.; et al. Knowledge Graph Neural Network with Spatial-Aware Capsule for Drug-Drug Interaction Prediction. *IEEE J. Biomed. Health Inform.* **2024**, *29*, 1771–1781.
214. Zhao, B.W.; Hu, L.; You, Z.H.; et al. HINGRL: Predicting Drug-Disease Associations with Graph Representation Learning on Heterogeneous Information Networks. *Brief. Bioinform.* **2022**, *23*, bbab515.
215. Zhang, R.; Wang, Z.; Wang, X.; et al. MHTAN-DTI: Metapath-Based Hierarchical Transformer and Attention Network for Drug-Target Interaction Prediction. *Brief. Bioinform.* **2023**, *24*, bbad079.

216. Li, D.; Yang, Y.; Cui, Z.; et al. LLM-DDI: Leveraging Large Language Models for Drug-Drug Interaction Prediction on Biomedical Knowledge Graph. *IEEE J. Biomed. Health Inform.* **2025**, *30*, 773–781.
217. Hua, Y.; Feng, Z.; Song, X.; et al. MMDG-DTI: Drug-Target Interaction Prediction via Multimodal Feature Fusion and Domain Generalization. *Pattern Recognit.* **2025**, *157*, 110887.
218. Liu, J.; Guan, S.; Zou, Q.; et al. AMDGT: Attention Aware Multi-Modal Fusion Using a Dual Graph Transformer for Drug-Disease Associations Prediction. *Knowl. Based Syst.* **2024**, *284*, 111329.
219. Li, D.; Zhao, F.; Yang, Y.; et al. Multi-View Contrastive Learning for Drug-Drug Interaction Event Prediction. *IEEE J. Biomed. Health Inform.* **2025**. https://doi.org/10.1109/JBHI.2025.3600045.
220. Zhang, P.; Che, C.; Jin, B.; et al. NCH-DDA: Neighborhood Contrastive Learning Heterogeneous Network for Drug-Disease Association Prediction. *Expert Syst. Appl.* **2024**, *238*, 121855.
221. Hu, L.; Pan, X.; Yan, H.; et al. Exploiting Higher-Order Patterns for Community Detection in Attributed Graphs. *Integr. Comput. Aided Eng.* **2021**, *28*, 207–218.
222. Yang, Y.; Li, G.; Li, D.; et al. Integrating Fuzzy Clustering and Graph Convolution Network to Accurately Identify Clusters from Attributed Graph. *IEEE Trans. Netw. Sci. Eng.* **2025**, *12*, 1112–1125.
223. Newman, M.E.; Girvan, M. Finding and Evaluating Community Structure in Networks. *Phys. Rev. E* **2004**, *69*, 026113.
224. Yang, Y.; Su, X.; Zhao, B.; et al. Fuzzy-Based Deep Attributed Graph Clustering. *IEEE Trans. Fuzzy Syst.* **2023**, *32*, 1951–1964.
225. Kempe, D.; Kleinberg, J.; Tardos, É. Maximizing the Spread of Influence through a Social Network. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 137–146.
226. Cohen, E.; Delling, D.; Pajor, T.; et al. Sketch-Based Influence Maximization and Computation: Scaling up with Guarantees. In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 629–638.
227. Ma, J.; Gao, W.; Wong, K.F. Rumor Detection on Twitter with Tree-Structured Recursive Neural Networks. In Proceedings of the Association for Computational Linguistics, Melbourne, Australia, 15–18 July 2018.
228. Khoo, L.M.S.; Chieu, H.L.; Qian, Z.; et al. Interpretable Rumor Detection in Microblogs by Attending to User Interactions. In Proceedings of the 2020 AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020, Volume 34; pp. 8783–8790.
229. Fan, W.; Ma, Y.; Li, Q.; et al. Graph Neural Networks for Social Recommendation. In Proceedings of the 2019 The World Wide Web Conference, San Francisco, CA, USA,13–17 May 2019; pp. 417–426.
230. Wu, J.; Wang, X.; Feng, F.; et al. Self-Supervised Graph Learning for Recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Montreal, QC, Canada 11–15 July 2021; pp. 726–735.
231. Albert, R.; Jeong, H.; Barabási, A.L. Error and Attack Tolerance of Complex Networks. *Nature* **2000**, *406*, 378–382.
232. Yu, B.; Yin, H.; Zhu, Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *arXiv* **2017**, arXiv:1709.04875.
233. Talebpour, A.; Mahmassani, H.S. Influence of Connected and Autonomous Vehicles on Traffic Flow Stability and Throughput. *Transp. Res. Part C Emerg. Technol.* **2016**, *71*, 143–163.
234. Aleta, A.; Meloni, S.; Moreno, Y. A Multilayer Perspective for the Analysis of Urban Transportation Systems. *Sci. Rep.* **2017**, *7*, 44359.
235. Motter, A.E.; Lai, Y.C. Cascade-Based Attacks on Complex Networks. *Phys. Rev. E* **2002**, *66*, 065102.
236. Liao, W.; Bak-Jensen, B.; Pillai, J.R.; et al. A Review of Graph Neural Networks and Their Applications in Power Systems. *J. Mod. Power Syst. Clean Energy* **2021**, *10*, 345–360.
237. Vázquez-Canteli, J.R.; Nagy, Z. Reinforcement Learning for Demand Response: A Review of Algorithms and Modeling Techniques. *Appl. Energy* **2019**, *235*, 1072–1089.
238. Musleh, A.S.; Chen, G.; Dong, Z.Y. A Survey on the Detection Algorithms for False Data Injection Attacks in Smart Grids. *IEEE Trans. Smart Grid* **2019**, *11*, 2218–2234.
239. Buldyrev, S.V.; Parshani, R.; Paul, G.; et al. Catastrophic Cascade of Failures in Interdependent Networks. *Nature* **2010**, *464*, 1025–1028.
240. Tao, F.; Zhang, H.; Liu, A.; et al. Digital Twin in Industry: State-of-the-Art. *IEEE Trans. Ind. Inform.* **2018**, *15*, 2405–2415.

241. Thacker, S.; Adshead, D.; Fay, M.; et al. Infrastructure for Sustainable Development. *Nat. Sustain.* **2019**, *2*, 324–331.
242. Buchgeher, G.; Gabauer, D.; Martinez-Gil, J.; et al. Knowledge Graphs in Manufacturing and Production: A Systematic Literature Review. *IEEE Access* **2021**, *9*, 55537–55554.
243. Noy, N.; Gao, Y.; Jain, A.; et al. Industry-Scale Knowledge Graphs: Lessons and Challenges. *Queue* **2019**, *17*, 48–75.
244. Bao, J.; Guo, D.; Li, J.; et al. The Modelling and Operations for the Digital Twin in the Context of Manufacturing. *Enterp. Inf. Syst.* **2019**, *13*, 534–556.
245. Zhang, B.; Gai, J.; Du, Y.; et al. Beyond Weisfeiler-Lehman: A Quantitative Framework for GNN Expressiveness. *arXiv* **2024**, arXiv:2401.08514.
246. Zhang, B.; Fan, C.; Liu, S.; et al. The Expressive Power of Graph Neural Networks: A Survey. *IEEE Trans. Knowl. Data Eng.* **2024**, *37*, 1455–1474.
247. Li, Z.P.; Wang, S.G.; Zhang, Q.H.; et al. Graph Pooling for Graph-Level Representation Learning: A Survey. *Artif. Intell. Rev.* **2024**, *58*, 45.
248. Meng, L.; Li, M.; Yang, Y.; et al. GLPACO: Global and Local Perspective Adaptive Collaborative Optimisation for Graph Contrastive Learning. *Expert Syst. Appl.* **2025**, *305*, 130898.
249. Li, H.; Wang, X.; Zhang, Z.; et al. Out-of-Distribution Generalization on Graphs: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 10490–10512.
250. Wu, M.; Zheng, X.; Zhang, Q.; et al. Graph Learning Under Distribution Shifts: A Comprehensive Survey on Domain Adaptation, Out-of-Distribution, and Continual Learning. *arXiv* **2024**, arXiv:2402.16374.
251. Tian, Q.; Zhao, C.; Shao, M.; et al. MLDGG: Meta-Learning for Domain Generalization on Graphs. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Toronto, ON, Canada, 3–7 August 2025; Volume 1, pp. 1361–1372.
252. Li, H.; Zhang, Z.; Wang, X.; et al. Invariant Node Representation Learning under Distribution Shifts with Multiple Latent Environments. *ACM Trans. Inf. Syst.* **2023**, *42*, 1–30.
253. Ding, K.; Nouri, E.; Zheng, G.; et al. Toward Robust Graph Semi-Supervised Learning against Extreme Data Scarcity. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 11661–11670.
254. Cui, J.; Huang, W.; Wang, Y.; et al. Rethinking Weak Supervision in Helping Contrastive Learning. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 6448–6467.
255. Alchihabi, A.; Guo, Y. Learning Robust Graph Neural Networks with Limited Supervision. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Valencia, Spain, 25–27 April 2023; pp. 8723–8733.
256. Chen, N.; Liu, Z.; Hooi, B.; et al. Consistency Training with Learnable Data Augmentation for Graph Anomaly Detection with Limited Supervision. In Proceedings of the The Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
257. Qian, S.; Ying, H.; Hu, R.; et al. Robust Training of Graph Neural Networks via Noise Governance. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, Singapore, 27 February–3 March 2023; pp. 607–615.
258. Achiam, J.; Adler, S.; Agarwal, S.; et al. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.
259. Pan, S.; Luo, L.; Wang, Y.; et al. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 3580–3599.
260. Brown, T.; Mann, B.; Ryder, N.; et al. Language Models are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
261. Liu, Z.; Yu, X.; Fang, Y.; et al. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April–4 May 2023; pp. 417–428.
262. Zhang, C.; Zhang, C.; Zheng, S.; et al. A Complete Survey on Generative AI (AIGC): Is ChatGPT from GPT-4 to GPT-5 All You Need? *arXiv* **2023**, arXiv:2303.11717.
263. Zhang, H.; Wu, B.; Yuan, X.; et al. Trustworthy Graph Neural Networks: Aspects, Methods, and Trends. *Proc. IEEE* **2024**, *112*, 97–139.
264. Chen, A.; Rossi, R.A.; Park, N.; et al. Fairness-Aware Graph Neural Networks: A Survey. *ACM Trans. Knowl. Discov. Data* **2024**, *18*, 1–23.

265. Jia, J.; Ma, S.; Liu, Y.; et al. A Causality-Aligned Structure Rationalization Scheme Against Adversarial Biased Perturbations for Graph Neural Networks. *IEEE Trans. Inf. Forensics Secur.* **2023**, *19*, 59–73.
266. Wang, Z.; Zeng, Q.; Lin, W.; et al. Generating Diagnostic and Actionable Explanations for Fair Graph Neural Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, Pennyslvania, 25 February–4 March 2024; Volume 38; pp. 21690–21698.