A Survey on Graph Embedding Techniques for Biomedical Data: Methods and Applications

Yaozu Wu[a,1], Yankai Chen[b,*,1], Zhishuai Yin[c,*], Weiping Ding[d] and Irwin King[b]

[a]Foshan Xianhu Laboratory of the Advanced Energy Science and Technology Guangdong Laboratory, Foshan, China
[b]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong
[c]School of Automotive Engineering, Wuhan University of Technology, Wuhan 430070, China
[d]School of Information Science and Technology, Nantong University, Nantong 226019 China

ARTICLE INFO

Keywords:
Biomedical data
Biomedical graph
Graph embedding
Graph neural network
Graph contrastive learning
Healthcare

ABSTRACT

As a result of the expeditious advancement of biomedical technologies, a plethora of relational data linking biomedical entities such as genes, proteins, and drugs have been collected for modern biomedical research. Biomedical graphs, one of the most popular ways to represent relational data, can easily describe different complex biomedical systems, including molecular-level, multi-omics-level, therapeutics-level, and healthcare-level interactions. However, traditional graph analysis methods still suffer from two difficulties (i.e., heterogeneities and dynamic properties of biomedical graphs) when handling high-dimensional, multi-modal, and sparsely interconnected biomedical data. To address these issues, graph embedding methods that can effectively analyze biomedical graphs have received a significant amount of attention recently. Generally, graph-based data is converted into a low-dimensional vector space with its structural properties and well-reserved information. These vectorized representations are used for further computation in various downstream biomedical tasks, such as gene function prediction and drug-target interaction prediction. In this article, we focus on the application of graph data in the biomedical domain and mainly introduce recent developments of graph embedding techniques (homogeneous, heterogeneous, and dynamic graph embedding techniques), including methodologies and related biomedical tasks. We also summarize relevant biomedical datasets and open-source implementations. We further discuss existing limitations and potential solutions. We hope this survey can provide useful directions for researchers who are interested in using graph embedding methods to solve problems in the biomedical field.

1. Introduction

Recent developments in biomedical technology have led to a significant rise in relational data interconnecting biomedical entities, from molecular interactions (e.g., drugs and proteins, etc.) to healthcare systems (e.g., diseases and medical concepts, etc.). As shown in Figure 1, the relational biomedical data at different scales [97] (e.g., molecular-level data, multi-omics-level data, therapeutics-level data, and human healthcare-level data) can be naturally represented by the graph format, as graphs can well capture the relations (i.e., edges) between biomedical objects (i.e., nodes). For instance, based on the physical interactions of atoms, molecular structures (e.g., the structure of proteins and chemical compounds) can be presented in a molecular-level graph consisting of nodes (i.e., atoms or amino acids) and edges (i.e., chemical bonds or peptide bonds) [117]. To introduce another example, as the most common bio-entities, multi-omics data (including genomics, proteomics, and transcriptomics) are usually represented as biomedical graphs to study life mechanisms, such as gene-gene interactions graphs, protein-protein interactions graphs, and miRNA-disease associations graphs, etc [25, 168, 209]. Analyzing biomedical graphs provides a deeper understanding of interconnected knowledge behind the graph topology, which ultimately facilitates better clinical treatment for human healthcare. For better learning of biomedical graphs, we divided biomedical graphs into three types: homogeneous, heterogeneous, and dynamic graphs, mainly by following the conventional graph-based methodologies [161, 210, 97] in academia to focus on their fundamental graph properties. However, due to the graph heterogeneity (i.e., graphs contain multiple types of nodes and edges) and dynamic property (i.e., graph information is evolving over time) of biomedical graphs, early traditional methods usually require designing complicated mechanisms across various-typed and sparsely distributed data components, which may lead to the high dimensionality of data permutation and insufficient information retrieval. Thus, these methods in the biomedical domain suffer from expensive computation costs and unsatisfactory performance, making them hard to deploy for realistic large-scale biomedical data. Taking protein-protein interaction (PPI) analysis as an example, early algorithms have an exponential time complexity and a fast performance deterioration after they are applied in large-scale PPI graphs with millions of nodes [66].

To address these issues, **graph embedding** methods have received much attention recently [181, 12, 27, 6]. Specifically, graph embedding methods map graph nodes into low-dimensional embedding spaces while preserving structural properties. These learned embeddings can be further input as condensed features into the downstream machine learn-

*Corresponding author. [1]Equal contribution.
The main content of this work was done at CUHK.
yaozuwu279@gmail.com (Y. Wu); ykchen@cse.cuhk.edu.hk (Y. Chen); Zyin@whut.edu.cn (Z. Yin); dwp9988@163.com (W. Ding); king@cse.cuhk.edu.hk (I. King)
ORCID(s): 0000-0001-8106-6447 (I. King)

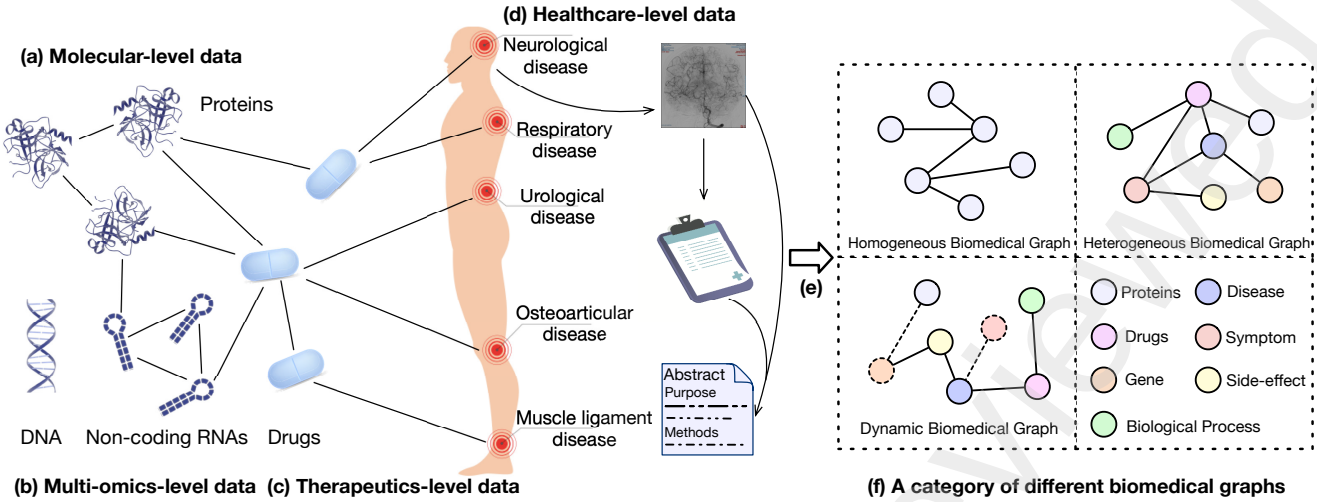

Figure 1: An overview of different biomedical data varying from molecular level to healthcare level, and a category of biomedical graphs. (a) Molecular-level data. (b) Multi-omics-level data. (c) Therapeutics-level data. (d) Healthcare-level data. (e) Generally, there are two ways to construct biomedical graphs. The first is based on manually curated databases where the data is usually derived from experimentally measured physical interactions, such as PPI and DrugBank, etc (as shown in Section 5). In addition, natural language processing techniques are applied to extract knowledge from different medical texts to construct biomedical graphs. (f) A category of different biomedical graphs. Based on the properties of nodes and edges, biomedical graphs can be categorized into three categories: homogeneous, heterogeneous, and dynamic biomedical graphs.

ing tasks, e.g., node classification, clustering, link prediction, and visualization, introducing enormous opportunities for biomedical data science (as shown in Figure 2). For example, graph embeddings are natural to be used in *biological entity clustering* that groups a set of objects with similar properties, introducing the narrowed search space for biological information exploration [192]. Based on such low-dimensional embeddings, downstream algorithms are usually more efficient than their counterparts that operate on the raw relational data. Consequently, a reduction of holistic computational overhead can be well achieved when solving biological computation problems. Moreover, compared with traditional algorithms focusing on local information, graph embedding-based algorithms are more capable of exploring high-order relational information among different biomedical objects, which thus greatly contributes to intelligent clinical decision making [237].

Related work. There exist several other related reviews on the topic of graph embedding in the biomedical domain that are summarized in Table 1. Su et al. [161] and Yi et al. [210] mainly elaborate graph embedding methods and graph neural networks for homogeneous and heterogeneous biomedical graphs. Other existing reviews [125, 222] discuss the applications of graph embedding methods in the biomedical fields. Authors in [128, 123] investigate biomedical knowledge graph embedding methods in various related tasks. The recent study [97] mainly discusses future directions of graph embedding methods for biomedical tasks.

However, these reviews are not comprehensive and systematic owing to the complexity of biomedical graphs (heterogeneity and dynamic property). In this study, our contribution consists of three main parts:

- **Comprehensive review.** We comprehensively review existing graph embedding models for different types of graphs, including homogeneous, heterogeneous, and dynamic graphs. We also investigate graph contrastive learning techniques, which are advantageous for addressing data quality issues in the biomedical field. In addition, we introduce related tasks in the biomedical domain from the perspective of common graph applications.
- **Accessible resources.** We organize the state-of-the-art models and collect their open-source implementations, and also attach related benchmarks from various real-life biomedical applications.
- **Discussion.** We discuss the major limitations of existing methods and several possible solutions.

2. Preliminaries

In this section, we introduce important concepts that are widely used in graph embedding methods.

Notations. In this paper, we use the bold lowercase characters, bold uppercase characters and calligraphy characters to denote the vectors, matrices and sets, respectively. Non-bold characters are used to denote graph nodes and edges. We introduce several important concepts as follows:

1. **Graph.** A graph G is represented as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes (or vertices), i.e., $\mathcal{V} = \{v_1, \dots, v_{|V|}\}$, and $\mathcal{E}$ denotes a set of edges connecting nodes, i.e., $\mathcal{E} = \{e_1, \dots, e_{|E|}\}$. Each edge E_k is a pair of nodes (v_i, v_j) with $v_i, v_j \in \mathcal{V}$. A graph is *undirected* means that, for each edge $(v_i, v_j) \in \mathcal{E}$, $(v_j, v_i) \in \mathcal{E}$. Otherwise, if the existence of edge $v_i, v_j \in \mathcal{E}$ does not necessarily imply $(v_j, v_i) \in \mathcal{E}$, the graph is *directed*.

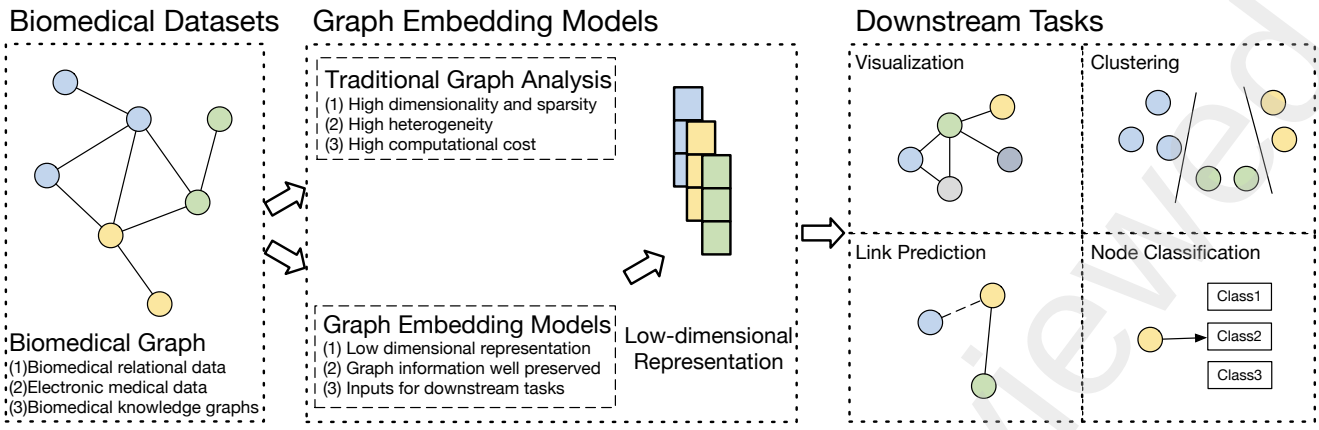

Figure 2: Comparison between traditional graph analysis methods and graph embedding methods.

Table 1
Existing work summary, where ✓ and – denote whether the technique is thoroughly or briefly introduced, N/A means that there is no description).

Papers		[161]	[125]	[222]	[128]	[123]	[210]	[97]
Year		2018	2019	2019	2020	2020	2021	2022
Graph types	Homogeneous grpah	✓	–	✓	–	N/A	✓	–
	Heterogeneous graph	✓	–	N/A	✓	✓	✓	–
	Dynamic graph	N/A	N/A	N/A	N/A	N/A	N/A	–
Applications	Node classification	✓	✓	✓	✓	✓	✓	✓
	Clustering	✓	✓	N/A	N/A	✓	N/A	✓
	Link prediction	✓	✓	✓	✓	✓	✓	✓
	Visualization	✓	✓	N/A	N/A	N/A	N/A	N/A
	Graph level application	N/A	N/A	N/A	N/A	N/A	✓	✓

In addition, let $\mathcal{A}$ denote the set of node types. There exists a mapping function $\Phi: \mathcal{V} \rightarrow \mathcal{A}$ associates each node with its corresponding node type. $\mathcal{R}$ denotes the set of edge types. Similarly, there is a function $\Psi: \mathcal{E} \rightarrow \mathcal{R}$ that maps each edge with the corresponding edge type.

2. **Homogeneous Graph.** A homogeneous graph $G = (\mathcal{V}, \mathcal{E})$ is a graph with only one type of nodes and one type of edges, i.e., $|\mathcal{A}| = |\mathcal{R}| = 1$.
3. **Heterogeneous Graph.** A heterogeneous graph $G = (\mathcal{V}, \mathcal{E})$ is graph that nodes or edges belong to different types, i.e., $|\mathcal{A}| + |\mathcal{R}| > 2$.
4. **Dynamic Graph.** In a dynamic graph, each node $v \in \mathcal{V}$ is associated with the tuple (t_1^v, t_2^v) where t_1^v, t_2^v are the start and end timestamps for the existence of v ($t_1^v \leq t_2^v$). Each edge $e \in \mathcal{E}$ is associated with the tuple (t_1^e, t_2^e) that t_1^e, t_2^e are respectively the start and end timestamps for the existence of the edge e ($t_1^e \leq t_2^e$).
5. **Adjacency Matrix.** Adjacency matrix $\boldsymbol{A}$ is a $n \times n$ matrix with $A_{ij} = 1$ or $A_{ij} = 0$ based on whether node i and node j are connected or not.
6. **Graph Laplacian Matrix.** An unnormalized graph Laplacian matrix is a $|V| \times |V|$ matrix given by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$ is the diagonal matrix denoting node degrees, i.e., D_{ii} is the degree of node v_i and $D_{ij} = 0$ for any $v_j \neq v_i$. The normalized graph Laplacian matrix is given by $\widetilde{\boldsymbol{L}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$, where $\boldsymbol{L}$ is the unnormalized graph Laplacian matrix.
7. **First Order Proximity.** The first order proximity between nodes v_i and v_j is the local pairwise proximity measure indicated by edge weight w_{ij}. Two nodes are more similar if they are connected with a larger w_{ij}.
8. **Second Order Proximity.** The second order proximity between nodes v_i and v_j is determined by the similarity of v_i's neighbourhood and v_j's neighbourhood. If v_i and v_j have many similar neighbors, the two nodes will have a high second-order proximity.
9. **Meta-path.** A meta-path is an ordered path that consists of node types that connects via edge types defined on the graph schema, which describes a composite relation between the types of nodes and edges involved. The usual format of a h-length meta-path is $v_1 \xrightarrow{r_1} a_2 \cdots \xrightarrow{r_{h-1}} v_h$, where v_i and r_i are node type and edge type.

Problem 1 (Graph Embedding). Given a graph $G = (\mathcal{V}, \mathcal{E})$, and a predefined embedding dimensionality d where $d \ll |V|$. As shown in Figure 3, graph embeddings map objects o in the graph into a d-dimensional space, where objects close to each other in the graph have similar latent representations. The objects can be nodes, edges and subgraphs in the graph.

3. Taxonomy of Graph Embedding Models

Different biomedical systems contain complex bio-entities and multiplex interactions from the molecular level to the clinical level, which could use multi-level biomedical graphs to describe. Following the conventional graph-based methodology, we organize biomedical graphs into three categories: homogeneous, heterogeneous, and dynamic graphs. In addition, based on different biomedical graphs, we review three

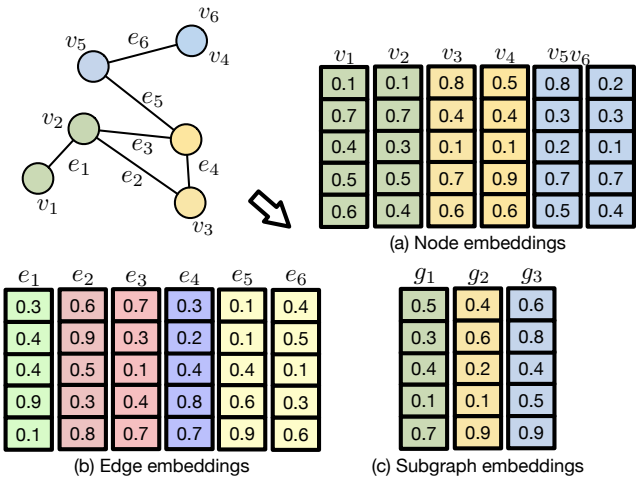

Figure 3: Node, edge and subgraph embeddings.

streams of corresponding graph embedding methods (as shown in Figure 4).

3.1. Homogeneous Graph Embedding Models

Generally, homogeneous graph embedding methods can be categorized into: (1) *matrix factorization-based*, (2) *random walk-based*, and (3) *deep graph learning-based*.

3.1.1. Matrix Factorization-based Methods

Inspired by the classical dimensionality reduction techniques, matrix factorization-based methods usually interpret the graph characteristics (e.g., node pairwise similarity) in a matrix form and decompose this matrix to attain the node embeddings [27]. The target of matrix factorization is to preserve the embedded graph characteristics as much as possible. There are two main types of matrix factorization methods for graph embeddings: *Node proximity matrix factorization* and *Graph laplacian matrix factorization*.

Node proximity matrix factorization. Node proximity matrix factorization-based methods usually approximate the node proximity into a low-dimensional embedding space. The objective of preserving the node proximity is to minimize the difference between the node proximity matrix and the approximation matrix as follows:

$$\min ||\boldsymbol{W} - \boldsymbol{X}\boldsymbol{C}^T||, \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}^{|V|\times|V|}$ is the node proximity matrix, $\boldsymbol{X} \in \mathbb{R}^{|V|\times d}$ is the node embedding matrix and $\boldsymbol{C} \in \mathbb{R}^{|V|\times d}$ is the embedding for the context nodes [13]. Following this objective, one classical solution is to use Singular Value Decomposition (SVD) [47] on $\boldsymbol{W}$. Recently, some other methods solve the objective in Equation (1), such as GF [3] and TADW [205]. Different from these methods that focus on undirected graphs, some other models, such as GrapRep [13] and Hope [136], are proposed to solve the matrix factorization on directed homogeneous graphs.

Graph laplacian matrix factorization. The basic assumption of graph Laplacian matrix factorization is that nodes that are close in the graph, measured by certain similarity functions, stay as close as possible in the embedding space measured by the predefined distance function. Thus to obtain good embeddings for graph nodes, the common operation is to give a penalty in the learning formulation if two nodes with higher similarity are erroneously embedded far apart. Formally, the optimal embedding $\boldsymbol{z}^*$ can be computed by using the objective function:

$$\boldsymbol{z}^* = \arg\min \boldsymbol{z}^T \boldsymbol{L}\boldsymbol{z}, \tag{2}$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ is the graph Laplacian matrix as we have introduced in Section 2. Usually, a constraint $\boldsymbol{z}^T\boldsymbol{D}\boldsymbol{z} = 1$ is given on Equation (2) to remove an arbitrary scaling factor in the embeddings [12]. Then Equation (2) is reduced to Equation (3) as follows:

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}^T\boldsymbol{D}\boldsymbol{z}=1} \boldsymbol{z}^T\boldsymbol{L}\boldsymbol{z} = \arg\min \frac{\boldsymbol{z}^T\boldsymbol{L}\boldsymbol{z}}{\boldsymbol{z}^T\boldsymbol{D}\boldsymbol{z}} = \arg\max \frac{\boldsymbol{z}^T\boldsymbol{W}\boldsymbol{z}}{\boldsymbol{z}^T\boldsymbol{D}\boldsymbol{z}}, \tag{3}$$

where the optimal $\boldsymbol{z}^*$ can be computed by finding the eigenvector that corresponds to the maximum eigenvalue of the eigenproblem $\boldsymbol{W}\boldsymbol{z} = \lambda\boldsymbol{D}\boldsymbol{z}$ [12].

There are many models based on graph Laplacian matrix factorization. The initial work Multi-dimensional scaling (MDS) [62] sets the Euclidean distance of two nodes' feature vectors as the similarity and minimizes the objective function in Equation (3) to learn the optimal embeddings. However, MDS mainly deals with linearly structured data in Euclidean space, but in real life, many data are non-linearly structured (i.e., graph data, tree-like data). To deal with the non-linear data, previous models such as IsoMAP [5], LE [4], and LLE [147] are proposed. Generally, they are non-linear dimensionality reduction techniques focusing on global geometry or local topology to find the best non-linear embeddings. AgLPP [75] introduces the concept of anchor graphs to significantly improve the efficiency of the previous matrix factorization model LPP [57]. In recent work, Yang et al. [206] define a new Weisfeiler-Lehman (WL) proximity matrix to capture the data dependence between edges and attributes. Based on the WL proximity matrix, BANE [206] formulates the matrix factorization for binary embedding learning, which achieves good informativeness of learned embeddings and computational efficiency from embedding binarization. In addition, DMCL [197], which hierarchically captures the semantic structures in the matrix factorization framework, performs non-negative factorization of data.

3.1.2. Random Walk-based Methods

Compared to matrix factorization-based methods, random walk-based methods are more adapted when the graph is partially observed. They can capture higher-order proximity between nodes. As illustrated in Figure 5, by sampling from the original graph, it is transformed to lists of random-walked paths. Then methods are applied to these sampled paths to learn the node embeddings by maximizing the occurrence probability.

One classical method is DeepWalk [143], which models a stream of short random walks and then invokes the Skip-Gram model to learn the node embeddings. Skip-Gram model maximizes the probability of predicting context nodes

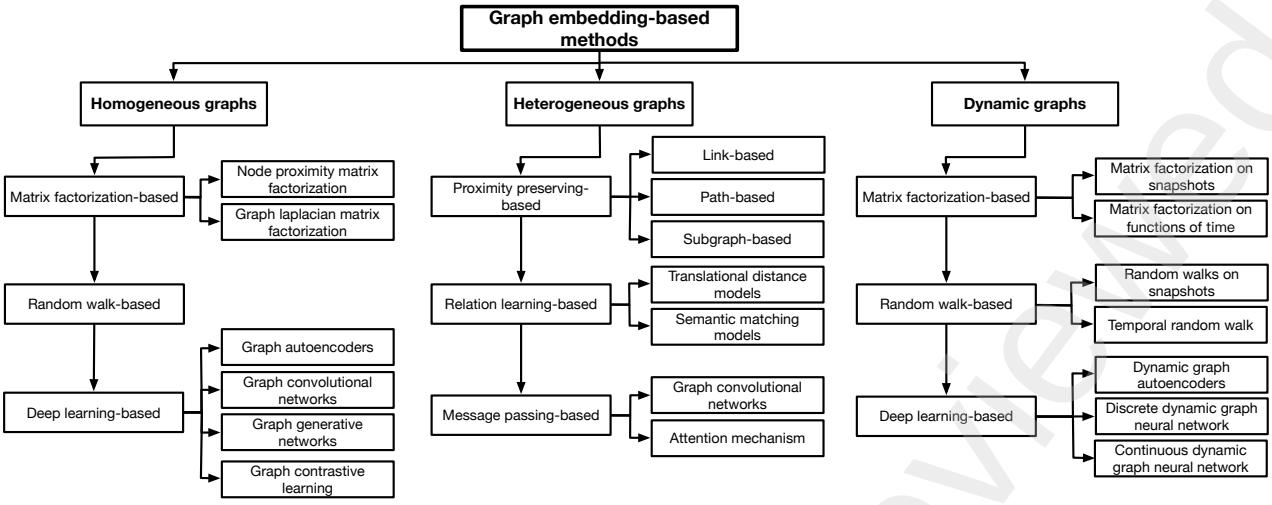

Figure 4: Taxonomy of graph embedding models.

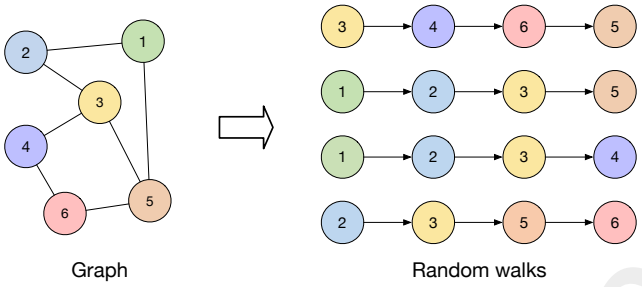

Figure 5: Random walk sampling process.

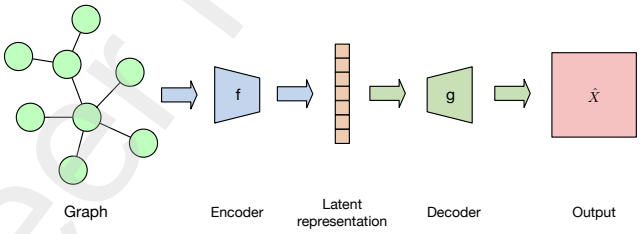

Figure 6: Architecture of autoencoder.

for each target node in a graph, namely the probability of target node's neighbourhoods [12]. It is achieved by the optimization problem:

$$\min_{y} -\log Pr\left(\left\{v_{i-k},\cdots,v_{i-1},v_{i+1},\cdots,v_{i+k}\right\} \mid \boldsymbol{x}_i\right), \quad (4)$$

where $\left\{v_{i-k},\cdots,v_{i-1},v_{i+1},\cdots,v_{i+k}\right\}$ are the observed nodes around node v_i within k window size, and $\boldsymbol{x}_i$ is the embedding of v_i [143]. Moreover, after making conditional independence assumption, the equation is further refined as:

$$\min_{y} -\log \sum_{-k\le j\le k} Pr\left(v_j \mid \boldsymbol{x}_i\right). \quad (5)$$

By computing truncated random walks on the graph, nodes that share similar context nodes in random walk paths will be as close as possible in the learned embedding space, as DeepWalk [143] can well capture the second-order (even higher-order) node proximity.

Some other methods, such as node2vec [51] and AWE [69] extend the sampling strategies to boost the performance further. Take the node2vec [51] model as an example, node2vec combines graph exploration via *breadth first search* (BFS) and *depth first search* (DFS) to perform the biased random walk on the graph. By doing so, node2vec can capture information from both local and global graph topology for effective embedding learning. Instead of alternating the sampling strategies, another research direction is to consider preserving additional proximity as well as the internal graph structures. For example, DeepCas [94] uses recurrent neural networks to embed information cascade paths. Moreover, considering node structure, node content, and node labels, TriDNR [138] uses a coupled neural network architecture to learn graph information. To avoid local minima caused by non-convex functions in both node2vec and DeepWalk, method HARP [16] proposes an effective multilevel graph representation learning paradigm. By recursively coalescing the input graph into smaller but structurally similar graphs, HARP learns a good initialization scheme from these small graphs. This equips HARP with a better initialization for node embeddings and thus provides a more stable training and convergence for embedding learning.

3.1.3. Deep Graph Learning-based Methods

Deep learning has shown outstanding performance in a wide variety of applications, ranging from acoustics, image processing to natural language processing [91]. Various deep architectures have been proposed to advance the graph analysis techniques significantly.

Graph autoencoders. As shown in Figure 6, graph autoencoders learn embeddings using an encoder to map nodes into a latent feature space, and a decoder decodes these latent embeddings to reconstruct the graph topological information. The idea of adopting the autoencoder for graph embedding is similar to node proximity matrix factorization, which minimizes the reconstruction error of the output and input.

Early work SDNE [179] applies a deep autoencoder to preserve the first and second-order node proximity, with the aim of capturing local and global graph structure. Specifically, the encoder maps the input adjacency matrix to an embedding space to produce node embeddings, and the decoder maps these embeddings to a reconstruction space to recover the original adjacency matrix. Another work DNGR [14] stacks a denoising autoencoder to encode embeddings and further decode the node similarity matrix via a probabilistic method named random surfing. Different from SDNE [179] and DNGR [14] that only consider the information of pairwise node connectivity, GAE, VGAE [82] and GC-MC [7] leverage graph convolutional networks (GCNs) to encode feature information and structural information of graph nodes. Details of GCN will be introduced later. As for the recent models, G2G [8] and DVNE [245] propose to encode graph nodes with a Gaussian distribution, producing better embedding learning for downstream tasks. Unlike previous work that uses Kullback-Leibler divergence [87] as the similarity measurement, DVNE uses Wasserstein distance as the similarity measure, which can simultaneously preserve the graph structure and model the uncertainty of nodes.

Graph generative networks. Graph generative network models aim to generate graphs based on the given observed subgraphs. In recent years, generative adversarial networks (GANs) have become an emerging stream for graph learning. Many researches incorporate generative adversarial methodology in graph embedding problems. For example, model ARVGA [137] not only minimizes the reconstruction errors of the graph structure but also enforces the latent representations to match a prior distribution. DGMG [102] implements a deep generative model to express the probabilistic dependencies among graph nodes and attributes. Another model NetRA [220] circumvents the requirement of an explicit prior distribution via encapsulating the joint inference in the adversarial training process. They learn node embeddings through jointly considering both the locality-preserving and global reconstruction constraints.

Graph convolutional networks. Recently, graph convolutional networks (GCNs) and variants have become popular. Generally speaking, there are two types of GCNs models, i.e., *spectral-based* GCNs and *spatial-based* GCNs. On one side, spectral GCNs are first introduced by Bruna et al. [11], which extends CNN algorithms from image processing to the graph domain. They investigate the graph Laplacian matrix and the main idea is similar to Fourier basis for signal processing [194]. On the other hand, spatial GCNs focus on local graph topologies and learn the node embeddings by exploring the nodes' multi-hop neighborhoods. Unlike spectral GCNs that load the whole graph matrices into the computation, which usually leads to large computational overhead, spatial GCNs are usually simple and easier to train.

Representative spectral GCNs [11, 83, 59, 33] adopt similar formulations. For example, method [83] introduces a well-behaved layer-wise propagation rule as follow:

$$\boldsymbol{H}^{(l+1)} = \sigma\Big(\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\Big), \tag{6}$$

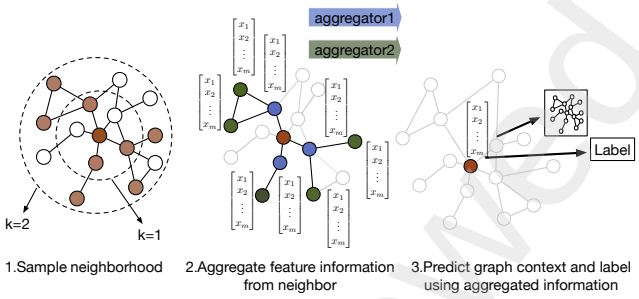

Figure 7: Illustration of GraphSAGE [52].

with $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix and $\hat{\boldsymbol{D}}$ is the diagonal node degree matrix of $\hat{\boldsymbol{A}}$. $\boldsymbol{W}^{(l)}$ is a weight matrix for the l-th neural network layer and $\sigma\langle\cdot\rangle$ is a non-linear activation function, e.g., *ReLU*. $\boldsymbol{H}^{(l)}$ is the input for layer l and $\boldsymbol{H}^{(l+1)}$ is the corresponding output. Then a GCN model with L layers can be constructed by inputing $\boldsymbol{H}^{(0)} = \boldsymbol{X}$ and applying the convolutional operations for L times.

Instead of considering graph convolutions from the perspective of the spectral domain, spatial-based methods define graph convolutions based on local graph structures. Concretely, spatial-based GCNs usually use *neighbor sampling* and *attention mechanism*. For example, as shown in Figure 7, GraphSAGE [52] samples fixed-size neighboring nodes and then conducts message passing via the connections with these sampled nodes. Concretely, the message propagation strategies can be described as:

$$\boldsymbol{H}_x^{(l+1)} = \sigma\Big(\Big[\boldsymbol{H}_x^{(l)}||\mathrm{AGG}(\{\boldsymbol{H}_i^{(l)}, \ \forall i \in \mathcal{N}(x)\})\Big]\Big), \tag{7}$$

where $\boldsymbol{H}_x^{(l+1)}$ is the embedding of node x in the $(l+1)$-th layer of model learning. AGG is the aggregation function proposed in [52] and $[\cdot||\cdot]$ is the concatenation operation between two embeddings. $\mathcal{N}(x)$ is the sampled neighbors of node x with the size $|\mathcal{N}(x)|$.

To improve the model efficiency and apply to large-scale graphs, researchers have proposed a number of improved models. StoGCN [21] uses historical node representations as a control variate to reduce the receptive field size of a graph convolution to an arbitrarily small scale. LGCN [43] designs the GCN model architecture specifically for large-scale graphs. Generally, LGCN learns the node information by effectively sampling subgraphs. FastGCN [18] interprets graph convolutions as integral transforms of node embedding functions under the probability measures. The method then applies the Monte Carlo approximation and variance reduction techniques to facilitate the efficient training process for large dense graphs. Cluster-GCN [23] samples node blocks from the dense subgraphs by using the graph clustering algorithm. Via restricting the neighborhood search over the sampled subgraph, Cluster-GCN can significantly improve the computational efficiency for handling larger graphs.

Another feature of spatial-based GCNs is *attention mechanism*. As shown in Figure 8, the representative method Graph Attention Network (GAT) [176] automatically determines the weights of node's neighbors when contributing in-

Table 2
Homogeneous graph embedding models.

Catagory	Sub-catagory	Methods	Description
Matrix factorization-based methods	Node proximity matrix factorization	[47, 3, 13, 205, 136]	Minimizing the difference between the node proximity matrix and the approximation matrix for reconstruction.
	Graph Laplacian matrix factorization	[4, 62, 147, 5, 57, 75, 206, 197]	Applying matrix factorization methods based on the graph Laplacian theory.
Random walk based methods		[143, 51, 138, 94, 69, 16]	Maximizing the probability of observing a node's neighborhood conditional on the node's embedding.
Deep learning-based methods	Graph autoencoders	[179, 14, 82, 7, 8, 245]	Using an encoder to map nodes into a latent feature space, and a decoder decodes these latent embeddings to reconstruct the graph's topological information.
	Graph generative networks	[137, 102, 220]	Incorporating generative methodology in graph embedding problems.
	Graph convolutional networks	[11, 83, 59, 33, 52, 21, 43, 18, 23, 176, 2, 230]	Applying a series of graph convolutional layers to aggregate information from neighboring nodes and updates the feature representations of each node.
	Graph contrastive learning	[177, 142, 55, 144, 252, 253, 214, 213, 172, 211, 77, 195, 92]	Comparing and updating node representations and encourage the network to produce representations that are similar for nodes that are close to each other in the graph and dissimilar for nodes that are far apart.

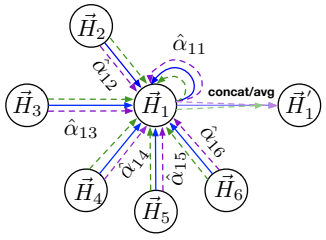

Figure 8: Illustration of GAT [176].

formation to this node. Consequently, GAT utilizes masked self-attentional layers to alleviate the disadvantages of prior methods. Specifically, as shown in Equation (8), GAT computes normalized coefficients, e.g., $\hat{\alpha}_{ij}$, using the softmax function across different neighbors as follows:

$$\hat{\alpha}_{ij} = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}(i)}\exp(e_{ik})}, \tag{8}$$

where $e_{ij} = f(\boldsymbol{H}_i, \boldsymbol{H}_j)$, f is the function to measure the attention between node embedding $\boldsymbol{H}_i$ and $\boldsymbol{H}_j$. To stabilize the learning process of self-attention, GAT uses multi-head attention to replicate K times of learning phases [176]. Then GAT outputs the feature-wise aggregation (typically by concatenating or adding), as shown in Equation (9), where $\hat{\alpha}_{ij}^k$ and $\boldsymbol{W}_k$ are the attention coefficient and the weight matrix specifying the linear transformation of the k-th replica, || represents concatenation.

$$\boldsymbol{H}_i = \Big|\Big|_{k=1}^{K}\sigma\Big(\sum_{j\in\mathcal{N}(j)}\widehat{\alpha_{ij}}\boldsymbol{W}^k\boldsymbol{H}_j\Big). \tag{9}$$

Recent work WYS [2] adopts graph neural networks and random walks to use differentiable attention weights when factorizing the co-occurrence matrix [194]. GaAN [230] also introduces the multi-head attention mechanism and adopts a specific self-attention mechanism that computes different weights for different heads.

Graph contrastive learning. Recent researchers have been attracted by the success of contrastive learning to tackle the real-world problems such as data sparsity issues. Introducing deep Infomax-based methods [61] into graph learning, DGI [177] maximizes the mutual information (MI) between node embeddings and a graph-level embedding for node classification tasks. Inspired by DGI [177], some other recent works follow such a strategy and try to get the agreement between different instances. InfoGraph [162] takes into account the mutual information regarding graph representations and sub-structural representations, such as nodes and subgraphs. Without explicit data augmentation, GMI [142] focuses on the MI between the input graph and embeddings of nodes and edges, respectively. To achieve a similar goal, MVGRL [55] uses graph diffusion kernels [84] to generate augmented graphs, which supplements the input graph with more global information. GCC [144] extracts different subgraphs from different input graphs and leverages contrastive learning to learn the intrinsic and transferable structural representations. GRACE [252] randomly removes edges and uses node features masking to generate augmented views of the input graph. GCA [253] samples sub-graph with the structure priors with node attributes randomly masked to generate the mutual information pairs. Moreover, GraphCL [214], JOAO [213] and DAGA [172] explores different types of graph augmentations in different scenes, such as node dropping, edge perturbation, attribute masking and subgraph. AutoGCL [211] employs an auto augmentation strategy to get

a set of learnable graph view generators. Moreover, AutoSSL [77] investigates how to automatically leverage multiple pretext tasks effectively. SimGRACE [195] uses the input graph and GNN model with its perturbed version as two encoders to obtain two correlated views. Furthermore, AF-GRL [92] focuses on nodes that share the local structural information and the global semantics with the graph to generate an alternative graph view.

3.2. Heterogeneous Graph Embedding Models

Heterogeneous graphs with different types of nodes and edges are appropriate to model rich semantics in real-world inter-related data. In fact, heterogeneous graphs are ubiquitous in many real-world scenarios, including bibliographic graphs, social graphs, and biomedical graphs [185]. However, the heterogeneity of graphs makes it challenging to preserve heterogeneous semantics in their low-dimensional graph embeddings. Different approaches that utilize graph structures and attributes have been proposed to deal with the issue. Generally, heterogeneous graph embedding methods can be organized into three subcategories: (1) *proximity preserving-based*, (2) *relation learning-based*, and (3) *message passing-based*.

3.2.1. Proximity Preserving Methods

Proximity preserving methodology has been extended to the field of heterogeneous graphs. By focusing on different graph structures, researchers [185] divide proximity preserving methods into: *link-based*, *path-based*, and *subgraph-based* methods. We introduce them as follows:

Link-based methods. Considering the edge as the structure information to be saved, model PTE [166] partitions the heterogeneous graph into multiple bipartite graphs and then employs LINE [167] to learn node embeddings for each bipartite graph. The node embeddings can be inferred by jointly optimizing the objective function:

$$\begin{aligned} \text{Loss} &= \sum_{r\in\mathcal{R}}\sum_{i,j\in\mathcal{V}} w_{ij}^{(k)} \log \frac{\exp\left(\boldsymbol{e}_i^T \boldsymbol{e}_j\right)}{\sum_{i'\in\mathcal{V}_{\phi(i)}} \exp\left(\boldsymbol{e}_{i'}^T \boldsymbol{e}_j\right)}, \\ &= \sum_{i,j\in\mathcal{V}} w_{ij} \log \frac{\exp\left(\boldsymbol{e}_i^T \boldsymbol{e}_j\right)}{\sum_{i'\in\mathcal{V}_{\phi(i)}} \exp\left(\boldsymbol{e}_{i'}^T \boldsymbol{e}_j\right)}, \end{aligned} \tag{10}$$

where $\mathcal{R}$ is the set of edge types, $w_{ij}^{(k)}$ is the type-k edge weight of (i, j) and $w_{ij} = \sum_k w_{ij}^{(k)}$ is the total edge weight.

Similar to PTE [166], authors in [198] propose a model called Embedding of Embedding (EOE) to learn embeddings for coupled heterogeneous graphs consisting of two different but related sub-graphs. EOE encodes both intra-graph and inter-graph edges. In addition, EOE also incorporates a harmonious embedding matrix for the embedding learning. PME [17] introduces a relation-specific embedding matrix to project nodes and relations into different latent spaces and then calculates the proximity between projected nodes to learn latent embeddings. MVE [145] preserves the view-specific proximity of nodes and then combines node representations in different views. To be more specific, MVE designs an attention mechanism to infer the weights of views for different nodes and then votes for robust node representations. Inspired by the methodology of GAN, HeGAN [65] captures the rich semantics on heterogeneous graphs and further trains a discriminator and a generator in a minimax formulation to generate robust latent representations, which is illustrated in Figure 9.

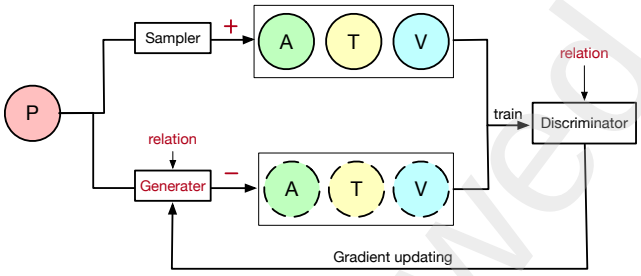

Figure 9: Architecture of HeGAN [65].

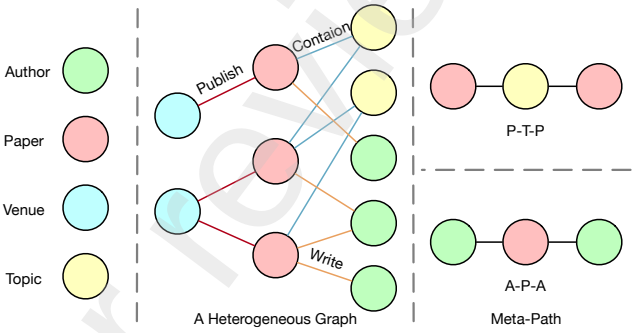

Figure 10: Example of meta-paths.

Recent models such as HEER [153] and AspEM [152] embed heterogeneous graphs via link representations. Concretely, HEER [153] builds edge embeddings atop node embeddings, which are equipped with inferred heterogeneous metrics for each edge type. The inferred metrics indicate the embedding dimensionality with important edge semantics. As a result, HEER updates node and edge embeddings by emphasizing different type-specific manifolds. Method AspEm [152] decomposes heterogeneous graphs into multiple pieces, and each piece is defined as a subgraph of the network schema. Then they propose an incompatibility measure to learn embeddings independently.

Path-based methods. Path-based methods using the concept of meta-paths can preserve high-order relations of heterogeneous graphs and thus capture more complex structural and semantic information. The definition of meth-paths can be found in Section 2. As shown in Figure 10, in a bibliographic graph with *authors*, *papers*, *venues* and *topics*, Author-Paper-Author (or APA) and Paper-Topic-Paper (or PTP) are two meta-paths that describe two relational paths with different semantic meanings between authors and papers. Thus, meta-paths can be defined to measure the high-order node proximity with specific semantics.

One representative work Metapath2vec [34] computes node embeddings by feeding meta-path-guided random walks

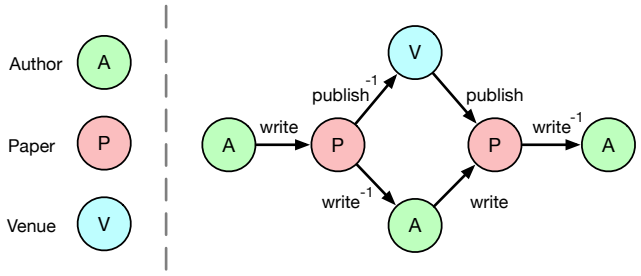

Figure 11: illustration of meta-graph [227].

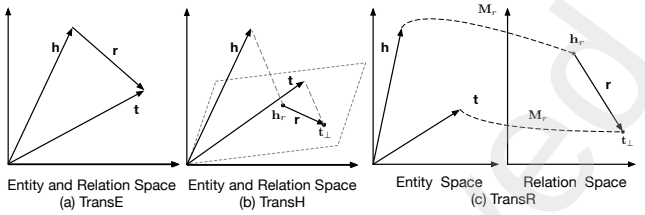

Figure 12: Illustration of TransE [10], TransH [190], TransR [106].

to the Skip-Gram [121] model. Inspired by the Metapath2vec model, a series of meta-paths-based methods have been proposed to carry out data mining tasks on the heterogeneous graphs. For example, HIN2vec [40] uses an approach based on the random walk and negative sampling to prepare training data in accordance with targeted relations specified in forms of meta-paths. It trains a neural network model by maximizing the likelihood of predicting relations to jointly learn the embeddings of nodes and edges. Recent work HeteSpaceyWalk [58] introduces a heterogeneous spacey personalized random walk by leveraging a non-Markovian spacey strategy. After getting the personalized random walk, it incorporates them into a Skip-Gram [121] model to learn the targeted embeddings. BHIN2vec [93] treats heterogeneous graph embedding as multiple relation-based tasks and uses an extended skip-gram technique to balance the various types of relations. BHIN2vec can balance the influence of different relations on node embeddings by adjusting the training ratio of different tasks. Moreover, model HHNE [189] extends metapath2vec [34] by conducting the meta-path-guided random walk in hyperbolic spaces, where the node similarity is measured by the hyperbolic geometry.

Subgraph-based methods. Compared to the single path, the subgraph has a more complex structure and thus can contain more structural and semantic information. Incorporating the subgraph into graph embedding algorithms can significantly improve the ability of capturing rich semantics between nodes in graph analysis learning. As shown in Figure 11, MetaGraph2Vec [227] proposes a new meta-graph concept, which contains multiple paths between nodes, to guide random walk generation in a heterogeneous graph. Then it utilizes the Skip-Gram model to learn latent representations of nodes and develops a heterogeneous negative sampling-based method that facilitates the efficient and accurate prediction of the node's heterogeneous neighborhood.

Recent work mg2vec [234] maps meta-graphs to the same embedding space as the nodes do and constrain node representations in a relationship-preserving manner. In addition, mg2vec uses both first- and second-order meta-graph embeddings to model not only the pairwise interactions among graph nodes, but also the individual node preference to further improve the model capability.

3.2.2. Relation Learning Methods

Translational distance models. To model the graph heterogeneity, relation learning models view a heterogeneous graph (a.k.a knowledge graphs) as a set of triplets (h, r, t). Each triplet is represented as a head entity (a.k.a graph node in geometric form) h having a relation r with tail entity t, e.g., (*Jeff Bezos, Founder of, Amazon*). The relation learning model generally face three questions to consider [30]: (1) how to define the representation form of entities and relations, (2) how to define the scoring function that measures the triplet plausibility, and (3) how to solve the optimization problem that maximizes the global plausibility of the existing triplets.

Motivated by translation invariance phenomena [122] in the word embedding space, i.e., $\boldsymbol{h}_{king} - \boldsymbol{t}_{queen} \approx \boldsymbol{h}_{man} - \boldsymbol{t}_{woman}$, model TransE [10] assumes that for each triplet (h, r, t), the embeddings of entities $\boldsymbol{h}$ and $\boldsymbol{t}$ are connected by the translation embedding vector $\boldsymbol{r}$. In other word, when both entities and relations are projected to the same low-dimensional embedding space $\mathbb{R}^d$, TransE follows a geometric principle: $\boldsymbol{h} + \boldsymbol{r} \approx t$. The triplet plausibility is computed via a score function as follow:

$$f_r(h, t) = ||\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}||_2^2. \tag{11}$$

TransE minimizes a margin-based ranking loss function as.

$$\sum_{(h,r,t)\in\mathcal{S}} \sum_{(h',r,t')\in\mathcal{S}'} \max(0, f_r(h, t) - f_r(h', t') + margin), \tag{12}$$

where $\mathcal{S}$ is the positive set containing the true fact, i.e., (h, r, t), and $\mathcal{S}'$ is the negative set of false triplets, i.e., (h', r, t'), that are not observed yet in the existing knowledge graphs.

Despite the good performance of TransE in prediction, it may not work well in dealing with mapping certain properties of relations, such as reflexive, one-to-many, many-to-one, and many-to-many [190]. To address these issues, authors in [190] develop the TransH model. TransH enables the entity to play different roles in different relations, by allowing each relation r to correspond to its relation-specific hyperplane. In other words, an entity would have different embedding after projecting into different relation-specific hyperplanes. For example, given the triplet (h, r, t), the entity embeddings $\boldsymbol{h}$ and $\boldsymbol{t}$ are firstly projected to the relation-specific hyperplane as follows:

$$\boldsymbol{h}_\perp = \boldsymbol{h} - \boldsymbol{W}_r^\top \boldsymbol{h} \boldsymbol{W}_r, \quad \boldsymbol{t}_\perp = \boldsymbol{t} - \boldsymbol{W}_r^\top \boldsymbol{t} \boldsymbol{W}_r, \tag{13}$$

where $\boldsymbol{W}_r$ is a relation-specific matrix and $\boldsymbol{h}_\perp, \boldsymbol{t}_\perp$ are the embedding projection of $\boldsymbol{h}$ and $\boldsymbol{t}$. Then the projections are connected by translation vector $\boldsymbol{r}$ which is the same as TransE. The score function of transH is defined as follows:

$$f_{\boldsymbol{r}}(h, t) = \left\|\boldsymbol{h}_\perp + \boldsymbol{d}_r - \boldsymbol{t}_\perp\right\|_2^2. \tag{14}$$

Following the TransE architecture, there are many extensions proposed. TransR [106] improves the TransH algorithm by adapting the original relation-specific hyperplane spaces to the relation-specific spaces. Method TransD [72] actually simplifies the learning design by substituting the projection matrix with a dynamic mapping matrix. To overcome the heterogeneity, later model TranSparse [73] proposes two models, i.e., TranSparse (share) and TranSparse (separate). The former model uses the adaptive sparse matrices to replace dense projection matrices for each relation r, and the latter one employs two separate spaces mapping matrices for each relation.

Semantic matching models. In contrast to translational distance models that measure the fact plausibility as the distance between two entities, semantic matching models measure the plausibility by matching latent semantics of entities and relations in the embedding space. The representative model RESCAL [130] embeds each entity with a vector to capture its latent semantics and each relation with a matrix to model pairwise interactions between latent embeddings. The score function is defined as follows:

$$f_r(h,t) = \boldsymbol{h}^T \boldsymbol{M_r} \boldsymbol{t}, \tag{15}$$

where $\boldsymbol{h}$, $\boldsymbol{t}$ are embeddings of the entities and $\boldsymbol{M_r}$ is a matrix associated with the relation. Later model HolE [129] composes head and tail entities by their circular correlation, achieving an advanced performance accordingly. There are several attempts to extend or simplify RESCAL. For instance, DistMult [204] restricts $\mathbf{M}_r$ as diagonal matrices to reduce the computation complexity. Model ComplEx [175] introduces complex-valued embeddings to better deal with the anti-symmetric relations. ANALOGY [108] focuses on the analogical properties of entities and relations to optimize latent representations. SimplE [80] presents an enhancement of Canonical Polyadic (CP) decomposition method [60], which enables the entity embedding to be learned dependently by considering the reverse of relations and calculates the average CP score of (h, r, t) and (t, r^{-1}, h) accordingly.

Recent work RotatE [164] defines each relation as a rotation (instead of a translation) from the source entity to the target entity in the complex vector space. RotatE is able to describe more complex relation patterns, including symmetry/asymmetry, inversion, and composition. Different from the standard complex space with a single real component and imaginary component, another work QuatE [233] computes node embedding vectors in the hypercomplex space with three imaginary components. It introduces the quaternion inner product and further proposes a new scoring function using the Hamilton product. The illustration of models RotatE and QuatE can be found in Figure 13. Another representative semantic matching method SME [9] first inputs embeddings of entities and relations in the input layer. As shown in Equation (16), the relation $\boldsymbol{r}$ is then combined with the head entity embedding $\boldsymbol{h}$ to obtain $g_{left}(h, r)$, and with

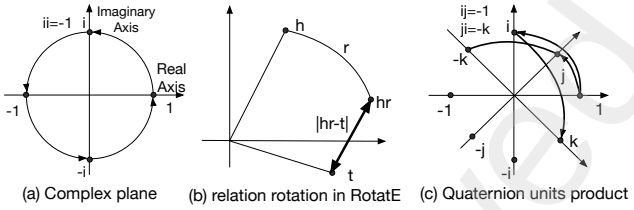

Figure 13: Illustration of RotatE and QuatE [164, 233].

the tail entity $\boldsymbol{t}$ to calculate $g_{right}(t, r)$ in the hidden layer.

$$\begin{aligned} g_{left}(h,r) &= \boldsymbol{M}_1 \boldsymbol{h} + \boldsymbol{M}_2 \boldsymbol{r} + b_h, \\ g_{right}(t,r) &= \boldsymbol{M}_3 \boldsymbol{t} + \boldsymbol{M}_4 \boldsymbol{r} + b_t. \end{aligned} \tag{16}$$

The score function is then defined to match $g_{left}(h, r)$ and $g_{right}(t, r)$ by their dot product as follows:

$$f_r(h,t) = g_{left}(h,r)^T g_{right}(t,r). \tag{17}$$

Inspired by contrastive learning, HaLE [180] designs a new learning objective based on query sampling to improve the performance and training speed of KGE models.

3.2.3. Message Passing Methods

Heterogeneous graph neural networks have been proposed to learn latent representations of heterogeneous graphs.

R-GCN [150] models relational data by using the GCN framework. Concretely, R-GCN uses multiple weight matrices $\boldsymbol{W}$ to project node embeddings into different relation spaces. The propagation rule to update of the node v_i is defined as:

$$\boldsymbol{h}_{v_i}^{(l+1)} = \sigma\Big(\sum_{r\in\mathcal{R}} \sum_{j\in\mathcal{N}_v^r} \frac{1}{c_{v,r}} \boldsymbol{W}_r^{(l)} \boldsymbol{h}_j^{(l)} + \boldsymbol{W}_0^{(l)} \boldsymbol{h}_v^{(l)} \Big), \tag{18}$$

where $\mathcal{N}_v^r$ denotes the neighbor set of node v under relation $r \in \mathcal{R}$. This indicates that R-GCN iteratively aggregates the neighbor information connected by the same edge type. $c_{v,r}$ is a problem-specific normalization constant that can either be learned or pre-defined.

HetGNN [226] applies a neighbors sampling strategy based on random walk with restart (RWR) to sample fixed-size node sets. It incorporates a graph neural network framework with two modules: (1) content aggregation and (2) neighbor aggregation. Concretely, content aggregation module extracts heterogeneous content (e.g., text or image) and computes the content embedding of v_i as follows:

$$f_1(v) = \frac{\sum_{i\in\mathcal{C}_v} \left[\overrightarrow{\text{LSTM}} \left\{ \mathbf{FC}\left(\boldsymbol{h}_i\right) \right\} \oplus \overleftarrow{\text{LSTM}} \left\{ \mathbf{FC}\left(\boldsymbol{h}_i\right) \right\} \right]}{|C_v|}, \tag{19}$$

where $\mathcal{C}_v$ is the heterogeneous contents of node v. $\boldsymbol{h}_i$ is the feature representation of i th content in $\mathcal{C}_v$. $\bigoplus$ denotes the concatenation operation. HetGNN uses a Bi-directional Long Short-Term Memory (Bi-LSTM) model to fuse the embeddings learned by the feature transformer $\mathbf{FC}$. The neighbor aggregation module aggregates content embeddings of a

Table 3
Heterogeneous graph embedding models.

Catagory	Sub-catagory	Methods	Description
Proximity preserving methods	Link-based methods	[166, 198, 145, 17, 152, 152, 153, 65]	Considering the edge as the structure information of heterogeneous graphs to be saved.
	Path-based methods	[34, 40, 93, 189, 58]	Using the concept of meta-paths to preserve high-order relations of heterogeneous graphs and thus capture more complex structural and semantic information.
	Subgraph-based methods	[227, 234]	Incorporating the subgraph into graph embedding algorithms to improve the ability of capturing rich semantics between nodes.
Relation learning methods	Translational distance models	[10, 190, 106, 72, 73]	Representing entities and relationships as vectors, and modeling the relationships between entities by using translations in vector space. If two entities are related, their representations should be translated versions of each other.
	Semantic matching models	[130, 204, 129, 175, 108, 81, 164, 233, 9, 180]	Measuring the plausibility by matching latent semantics of entities and relations in the embedding space.
Message passing methods	Graph convolutional networks	[150, 15, 226, 250, 223, 239, 41, 240]	Passing information along the edges of the graph, and updating the node representations based on the relationships and node features.
	Attention mechanism	[187, 64, 67]	Using attention mechanism to capture the most important relationships and nodes in the heterogeneous graph.
	Contrastive Learning	[156, 155, 188]	Minimizing a contrastive loss function that measures the difference between positive and negative examples. Positive examples are pairs of nodes that are connected by an edge in the graph, while negative examples are pairs of nodes that are not connected by an edge.

neighboring node group as:

$$f_2^r(v) = \frac{\sum_{v' \in \mathcal{N}_r(v)} \left[\overrightarrow{LSTM} \left\{ f_1\left(v'\right) \right\} \oplus \overleftarrow{LSTM} \left\{ f_1\left(v'\right) \right\} \right]}{\left| \mathcal{N}_r(v) \right|}, \quad (20)$$

here $\mathcal{N}_r(v)$ is the first-order neighbors of node v with relation r. $f_1\left(v'\right)$ is the content embedding of v' generated by content aggregation module. Then HetGNN further combines these embeddings with v's content embedding by considering the attention mechanism.

Recent work MV-ACM [240] characterizes each relation in a single viewpoint and utilizes the adversarial learning framework to learn the reciprocity between different relations. Authors in GTN [223] use a graph transformation layer to learn a soft selection of edge types and composite relations. This method generates new graph structures using multiple candidate adjacency matrices and learns node representations on the new graphs in an end-to-end fashion. MAGNN [41] comprehensively considered three main components to achieve state-of-the-art performance: (1) the node content transformation to encapsulate node attributes, (2) the intra-metapath aggregation to incorporate intermediate semantic nodes, and (3) the inter-meta path aggregation to combine messages from multiple meta-paths.

Methods described above usually partition the heterogeneous graph into multiple homogeneous subgraphs before learning their topological information. However, these approaches are ineffective in exploiting rich semantic associations with different types of edges in large-scale multi-relational graphs. For example, Cen et al. [15] propose a novel approach, GATNE, to capture the rich attributed information and utilize multiplex topological structures from different node types. RSHN [250] uses the original heterogeneous graph and defines coarsened line graphs to learn embeddings of nodes and edges without requiring any prior knowledge, such as meta-paths. Similarly, to get rid of the influence of meta-path selection on model learning, authors in NSHE [239] incorporate the network schema that comprehensively embraces the high-order structure to generate subgraphs. Then NSHE builds multi-task learning tasks to preserve the heterogeneous structure of each schema instance.

The great potential of the attention mechanism has been well demonstrated in recent work for heterogeneous graph embeddings. As shown in Figure 14, HAN [187] learns meta-path-oriented node embeddings from the different meta-path-based graphs that are converted from the original heterogeneous graph. Moreover, it leverages the attention mechanism to combine them into each node's representation. Model HetSANN [64] projects the node embeddings into the low-

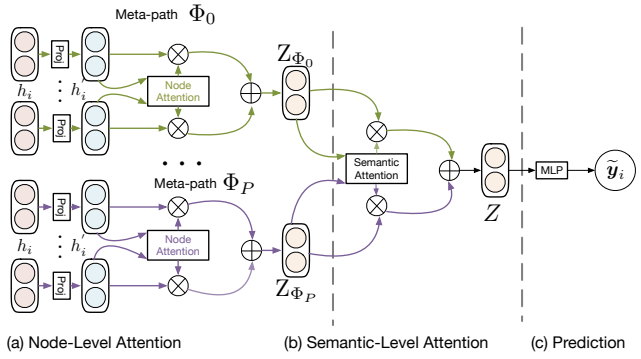

Figure 14: Overview of the HAN [187] model.

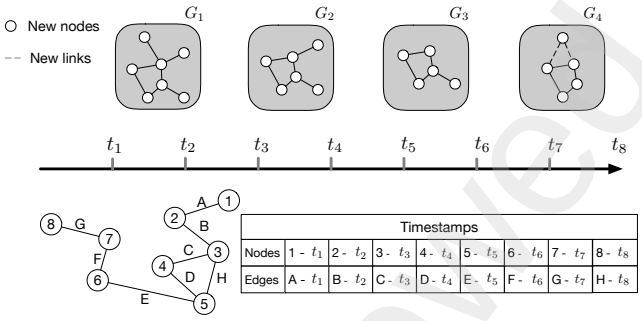

Figure 15: Illustration of a dynamic graph model [202].

dimensional entity spaces to model the transformation between heterogeneous nodes and applies the graph neural network to aggregate multi-relational information of node neighbors by means of the attention mechanism. The state-of-the-art model HGT [67] introduces an attention mechanism that designs node- and edge-type dependent parameters to characterize the heterogeneous attention over each edge.

For the model interpretability, JKT [156] proposes a joint GCN-based deep knowledge tracing framework to help capture high-level semantic information. Furthermore, Models Bi-CLKT [155] and HeCo [188] introduce a self-supervised learning framework into heterogeneous graph learning. Bi-CLKT [155] involves node-level contrastive learning and graph-level contrastive learning to obtain discriminative node representations and concept representations. Heco [188] employs cross-view contrastive learning and a view mask mechanism to learn high-level node embeddings.

3.3. Dynamic Graph Embedding Models

In the real world, graphs (including nodes, edges, and graph properties) evolve over time. For dynamic graph embedding problems, the latent representations need to maintain the structural relationships between nodes in the embedding space and capture the temporal information of graphs in the evolution. Typically, dynamic graph embedding methods can be categorized into: (1) *matrix factorization-based*, (2) *random walk-based*, and (3) *deep learning-based*. Generally, there are two main manners to model temporal information: *discrete-time manner* and *continuous-time manner*. Discrete-time manner represents a dynamic graph $\boldsymbol{G}$ as a sequence of graph snapshots $\boldsymbol{G} = \{\boldsymbol{G}_1, \ldots, \boldsymbol{G}_T\}$ within a given time interval, where $\boldsymbol{G}_k$ is a static graph with timestamp t_k. Continuous-time manner uses multiple timestamps to mark each edge. When an edge changes, new edges are created and annotated with timestamps. All nodes update their own timestamps when they are created, or their properties are changed.

3.3.1. Matrix Factorization-based Methods

Dynamic graph embedding methods based on matrix factorization express the evolutionary structure of graphs in the form of matrices, and differ in how to insert temporal dependencies in the matrix factorization process.

Authors in [95] propose a dynamic attributed graph embedding framework, namely DANE. Technically, it first applies the graph Laplacian eigenmaps factorization method to get the initial embedding $\boldsymbol{z}^t$ of the graph at the time $t = 1$. To capture temporal information, DANE applies matrix perturbation theory [158] to update the graph embedding at the following snapshot t. Similar to DANE [95], DHPE [246] performs generalized singular value decomposition (SVD) to get initial graph embedding at time t and uses matrix perturbation to update embedding results at later time $t + \delta t$. However, these perturbation methods will accumulate errors and may not be very effective if the graph evolves intensely over time. To solve this issue, TIMERS [236] optimally sets a restart time of SVD to reduce error accumulation. By setting a threshold, SVD model restarts automatically when the residual exceeds the rated threshold. In addition, instead of using first-order matrix perturbation theory to update the temporal properties of the dynamic graph, authors in [247, 38] assume that the embedding space is *time-smoothing*. In other words, for each node, the embeddings of two consecutive snapshots are similar, and the loss function term is used to obtain the best low-dimensional representations by minimizing the differences between the learned information from two consecutive snapshots.

Instead of considering the topological and temporal properties of dynamic graphs separately, another research direction obtains embeddings by characterizing the structure of the graph as a function of time. TMF [218] proposes a time-dependent similarity matrix $\boldsymbol{S}_t$ and decomposes it to learn the low-dimensional representations. The whole process can be abstracted with the following formulation:

$$\boldsymbol{S}_t = f\left(\boldsymbol{X}\boldsymbol{C}(t)^T\right), \tag{21}$$

where $\boldsymbol{X}$ is a constant matrix and $\boldsymbol{C}(t)$ is a time-dependent matrix and $f(\cdot)$ is an element-wise function.

3.3.2. Random Walk-based Methods

To generalize random walk-based methods for dynamic graphs, models need to not only generate sequences that capture topological structures but also capture temporal information. Based on the strategy of extracting temporal information from dynamic graphs, random walk-based methods can be divided into two different subcategories: (1) *Random walks on snapshots* and (2) *Temporal random walk methods*.

Random walks on snapshots. Considering a dynamic graph as the discrete-time manner, models perform random

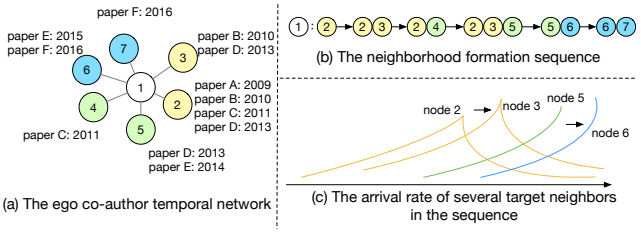

Figure 16: Neighborhood formation sequence [257].

walk-based methods on each snapshot of dynamic graphs and then use certain operations to combine time dependencies or update evolving nodes to obtain embeddings. Related work [32, 154, 244] applies node2vec [51] for each graph snapshot to get node embeddings at different time points. To fully grasp dynamic graph properties, [32] apply vector concatenation to node representations over time. Later model tNodeEmbed [154] optimizes the LSTM for specific tasks to get a final embedding of nodes. Furthermore, DynSEM [244] projects node embeddings into a common space using orthogonal procrustes and designs a joint loss function to learn temporal embeddings for all time snapshots.

In the case that only a few nodes and edges change over time, computing node embeddings for each snapshot can be computationally expensive. To address this issue, model Dynnode2vec [116] first generates embeddings for the initial timestamp by using the node2vec [51] model and only updates the evolved node embeddings instead of considering all nodes at the current timestamps. Work DNE [35] extends the LINE [167] model to the dynamic graph embedding framework, where only a part of nodes are iteratively updated, and the embeddings in each iteration have strong stability compared to retraining the whole graphs.

Temporal random walk methods. Different from the above methods based on discrete graph snapshots, Authors in CTDNE [126] model the dynamic graph in a continuous-time manner. They first perform a temporal random walk on the dynamic graph and then regard the task of learning the time-preserving node embeddings as an optimization problem:

$$\max_{f_d} \log \Pr\left(W_T = \left\{v_{i-q}, \cdots, v_{i+q}\right\} \setminus v_i \mid f_d\left(v_i\right)\right), \quad (22)$$

where f_d is the node embedding function, q is the context window size for optimization, and $W_T = \left\{v_{i-q}, \cdots, v_{i+q}\right\}$ is an arbitrary temporal context window $W_T \subseteq S_t$. Following CTDNE's paradigm, T-EDGE [104] develops to incorporate both temporal and weighted information of graphs to get the latent graph embeddings. Moreover, model HTNE [257] introduces the concept of neighborhood formation sequence (as shown in Figure 16) to describe the node evolution for embedding learning. Then it applies Hawkes Process [56] to model the neighborhood formation, with the assumption that the nodes with new arriving edges are impacted by historical events.

3.3.3. Deep Learning-based Methods

Deep learning methodology can also be applied for the dynamic graphs. Similarly, we categorize the deep learning-based methods into three subcategories: (1) *dynamic graph autoencoders*, (2) *discrete dynamic graph neural network (discrete DGNN)* and (3) *continuous dynamic graph neural network (continuous DGNN)*.

Dynamic graph autoencoders. For graph autoencoder-based approaches, Early work DynGEM [50] applies a deep autoencoder to generate highly nonlinear embeddings. In particular, to learn the embedding for the snapshot at time t, it uses a transfer learning paradigm that shares parameters between the two autoencoders for the snapshots at time $t-1$ and t. It proposes a heuristic method to widen the neural network layers at each time step and insert new layers which handle the problem of a growing number of graph nodes. Following DynGEM [50] architecture, LDANE [191] adds a margin-based ranking loss term in the loss function to regularize node's attributes information in dynamic graphs. In addition, to learn longer-term historical information, unlike DynGEM [50] framework, which only captures information from the previous step and ignores rich historical information, dyngraph2vec [49] takes the last l snapshots in the encoding as input to predict the following $(t+l+1)$ graph embeddings. They implement three different architectures: (1) dyngraph2vecAE, (2) dyngraph2vecRNN, and (3) dyngraph2vecAERNN, to embed historical information. Another work E-LSTM-D [20] runs the stacked LSTM module on the encoded hidden vectors to learn temporal dependencies. Instead of considering graphs as input, model NetWalk [219] first extracts a number of random walks from the dynamic graph to learn the node representations. By hybridizing local walks with a hidden layer of a deep autoencoder, this model is able to reconstruct the original graph with a less loss. The learning process utilizes a reservoir sampling strategy [219] to apply dynamic changes.

Discrete dynamic graph neural network. Discrete dynamic graph neural networks capture each graph snapshot and combine it with certain (deep) time series modules to jointly learn node embeddings. For example, RgCNN [124] and DyGGNN [165] use a standard LSTM module to get the time dynamics of the underlying graphs. The former model applies the PATCHY-SAN solution [131] to learn the spatial structure of graphs, and the latter one employs Gated Graph Neural Networks (GGNNs) for each snapshot to preserve the topology of dynamic graphs. Furthermore, EvolveGCN [139] applies a recurrent mechanism to capture the graph dynamic information and GC-LSTM model [19] learns the temporal features of the dynamic graphs also by integrating the LSTM module. In addition, *Attention Mechanism* is another useful mechanism for discrete DGNN to boost performance. The representative method is DySAT [148]. It applies GAT [176] for each snapshot to generate a sequence of node embeddings and uses the temporal attention block to learn the final representation. Instead of using self-attention, DyHAN [208] uses hierarchical attention to learn node embeddings, which leads to better performance in the link prediction task ac-

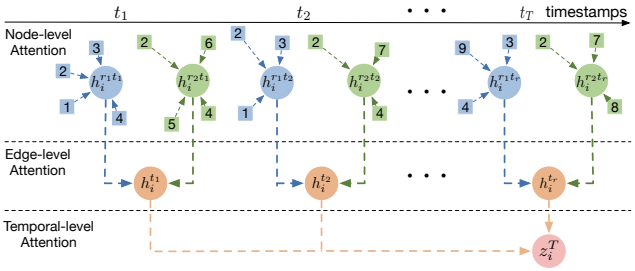

Figure 17: Overview of the DyHAN [208] model.

cordingly, as shown in Figure 17.

Continuous dynamic graph neural network. Modeling dynamic graphs in the continuous-time manner, KnowEvolve [173] proposes a deep recurrent framework to model the knowledge evolving processes in the form of interaction graphs. Via modifying an RNN model, it models the occurrence of a fact as a multidimensional temporal point process whose conditional intensity function is modulated by the relationship score for that fact [173]. Authors in DyRep [174] propose a recurrent deep representation learning framework that bridges two observed processes, namely the association process and communication process. The association process is used to deal with the change of graph topology, and the communication process is used to handle the other graph dynamics. As illustrated in Figure 18, JODIE [88] builds up a coupled RNN architecture and adds an attention layer to get final representations for graph nodes. Recent work DyGNN [115] consists of two components: (1) the update component and (2) the propagation component. The former component updates new interaction information of graph nodes, and the latter one propagates the updated information to the involved nodes' neighbors. DyGNN captures the sequential information of edges and the time intervals between edges coherently to keep the node information updated.

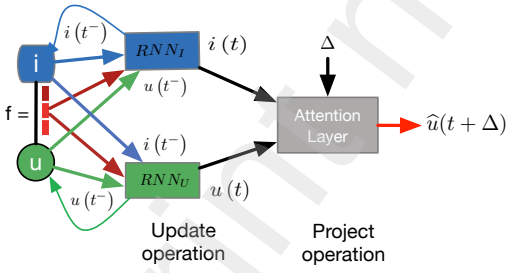

Figure 18: The JODIE model illustration [88].

4. Applications in Biomedical Domain

Graph embedding techniques can be applied to biomedical tasks, including *Pharmaceutical data analysis*, *Multi-omics data analysis*, and *Clinical data analysis* [161]. For instance, graph embeddings are optimized for specific biomedical **link prediction** tasks, such as drug-target interaction, drug-disease interaction, and drug-drug interaction prediction [169, 221, 254]. On the other hand, **node classification** is a crucial application used for intelligent diagnosis and triage in clinical data analysis [193]. In this section, we review some important tasks in the biomedical field from the perspective of common graph applications, including *node classification*, *clustering*, *link prediction*, *visualization*, *graph level applications*, and *graph information summarization*.

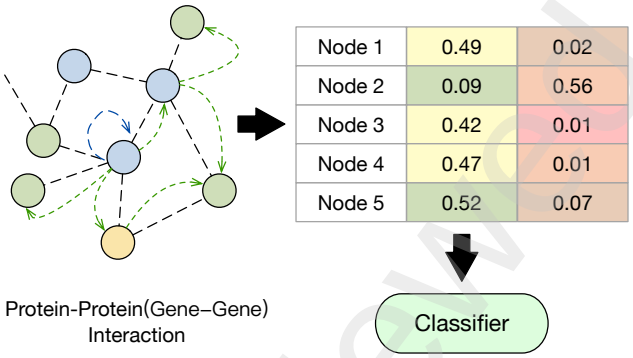

Figure 19: Node classification application. Given the gene-gene interaction (protein-protein interaction) network and other complementary information, researchers use the graph embedding model to get the low-dimensional representation of gene nodes (protein nodes), which can be used to learn the properties of genes or proteins.

4.1. Node Classification

There is a large gap between the number of discovered biomedical entities and the number of functionally annotated ones, so the annotation of biomedical information is a major challenge for bioinformatics. Nevertheless, exhaustively functional annotation of genes and proteins via large-scale experiments is often expensive and time-consuming [99]. Therefore, node classification, which predicts the attributes of unlabelled nodes, has become a crucial graph application to address this gap.

Gene function prediction. Inferring gene functions has become a representative work in biological graph analysis. ClusDCA [182] applies DCA to gene-gene interactions and gene ontology to learn low-dimensional representations for gene function prediction. In addition, NMFGO [216] conducts a non-negative matrix factorization method on a gene-term association matrix and then employs a semantic similarity-based nearest neighbor classifier to predict gene functions. In recent works, DeepMNE-CNN [141] designs a novel semi-supervised autoencoder method to integrate multiple graphs and generate low-dimensional feature representations. Then based on these integrated feature embeddings, it utilizes convolutional neural networks to annotate unlabeled gene functions. Li et al. [99] design a multi-view graph embedding method to learn low-dimensional representations among multiple interaction networks for the gene function prediction task.

Protein function prediction. Protein function prediction is another area that has received increasing attention in recent years. Having a comprehensive understanding of protein function can aid in solving a variety of biological problems, such as understanding disease mechanisms or finding drug targets.

Table 4
Dynamic graph embedding models.

Catagory	Sub-catagory	Methods	Description
Matrix factorization-based methods	Matrix factorization on snapshots	[247, 95, 218, 246]	Applying matrix factorization-based methods to get the initial embedding and using matrix perturbation to update embedding.
	Temporal matrix factorization	[236, 38]	Characterizing the structure of the graph as a function of time and decomposing the matrix to learn the low-dimensional representations.
Random walk-based methods	Random walks on snapshots	[32, 35, 116, 244, 154]	Considering a dynamic graph as the discrete-time manner, models perform random walk-based methods on each snapshot of dynamic graphs and then use certain operations to combine time dependencies or update evolving nodes to obtain embeddings.
	Temporal random walk methods	[126, 104, 257]	Performing a temporal random walk on the dynamic graph, and then regard the task of learning the time-preserving node embeddings as an optimization problem.
Deep learning-based methods	Dynamic graph autoencoders	[50, 191, 20, 219, 49]	Using a transfer learning paradigm that shares parameters between the two autoencoders for the snapshots at different times.
	Discrete dynamic graph neural networks	[124, 19, 148, 165, 139, 139, 208]	Capturing each graph snapshot and combine it with certain (deep) time series modules to jointly learn node embeddings.
	Continuous dynamic graph neural networks	[173, 88, 174, 115]	Using RNNs to maintain node embeddings in a continuous manner. A common characteristic of these models is that as soon as an event occurs or there is a change to the network, the embeddings of the interacting nodes are updated.

Yu et al. [217] apply downward random walks with restart on the gene ontology-directed acyclic graph to estimate the probability of missing functions. It is worth noting that the precise function of proteins often depends on their tissue, and different proteins will have disparate functions resulting from diverse tissues. Thus OhmNet [255] incorporates knowledge of a wide range of human tissues into PPI graphs. It proposes a hierarchical-aware node feature learning method based on node2vec [51] to represent the embeddings of proteins as low-dimensional vectors for protein function prediction. DeepNF [45] constructs low-dimensional representations based on a multi-modal deep autoencoder. DeepGraphGO [212] proposes an end-to-end, multispecies graph neural network-based method that allows one single model to be trained for all species, making the most of both protein sequence and high-order protein graph information for protein function prediction.

Diseases classification As electronic health records (EHRs) and electronic medical records (EMRs) contain rich realistic clinical information, the EHRs or EMRs analysis is expected to promote clinical research and significantly assist doctors in making better clinical decisions, eventually benefiting both patients and caregivers.

Clinical Decision Support Systems (CDSS) can improve the quality of medication and patient care by extracting information from clinical data [237]. Early work uses deep learning to boost CDSS systems. For example, Rajkomar et al. [146] use deep neural networks on EHR data to predict the mortality of patients. However, due to the scarcity of available biomedical data, CDSS systems using classical methods usually do not produce a good performance in the existence of rare diseases. To solve this problem, researchers pay attention to graph-based methods for performance improvement. For example, Choi et al. [24] propose a graph attention-based framework that infuses information from external medical hierarchical ontologies to learn EHR representations. And ME2Vec [193] uses word2vec, GAT [176], and LINE [167] to learn better vectorized representations of the medical services, doctors and patients for diagnosis.

Recent works [110, 48, 44] use heterogeneous information graphs or knowledge graphs that preserve the semantics of different types of associations between entities for modeling clinical data and disease diagnosis. HSGNN [110] firstly normalizes the edges and divides a heterogeneous EHR graph into multiple homogeneous graphs and then uses an end-to-end model based on GNN to help human experts make medical decisions. Ge et al. [44] employ the knowledge-based embedding techniques and Self-Attention Mechanism to capture the relations between medications for more informative representations accordingly. Recent model SMR [48]

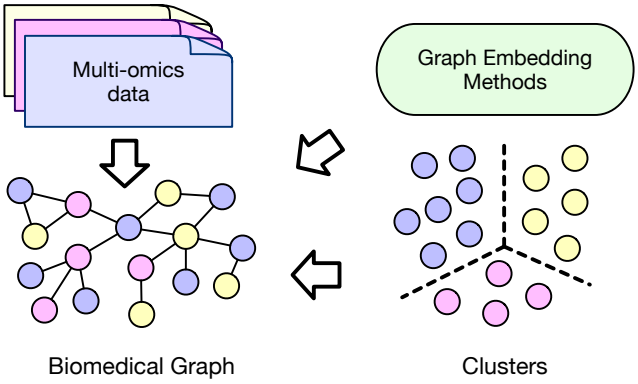

Figure 20: Clustering applications, where similar low-dimensional representations learned from biomedical graphs are grouped into clusters, play an essential role in biomedical graph analysis, such as single-cell RNA sequencing (scRNA-seq) data analysis, biomedical articles clustering, etc.

embeds diseases, medicines, patients, and relations into a shared low-dimensional space to learn node embeddings jointly.

4.2. Clustering

Identifying similarities and differences of data is another important task for large biomedical datasets. Clustering algorithms, which group similar objects into clusters so objects in the same set are more similar to each other than those in other groups, are particularly useful in biomedical graph analysis. The great potential of graph embeddings has been well demonstrated on clustering problems. Latent representations learned by graph embedding methods can show the relational knowledge of the biomedical graph in a low-dimensional space. It is more convenient to apply generic clustering algorithms by using these learned node embeddings (e.g., k-means), providing an open-ended alternative to traditional community detection techniques.

For example, document clustering, which extracts and visualizes complex relations inherent to scientific literature, has been a powerful approach to find and retrieve documents from rapidly growing datasets. Oniani et al. [135] construct a co-occurrence graph based on the COVID-19 Open Research Dataset (CORD-19) and get the representation of the derived co-occurrence graph using node2vec [51]. They further utilize the t-SNE algorithm to group nodes with similar embeddings into several discrete clusters. Similarly, Wise et al. [192] retrieve similar articles by encoding them into d-dimensional embeddings with the R-GCN [150] framework.

Furthermore, grouping cells is the focal of single-cell RNA sequencing (scRNA-seq) data analysis, which significantly deepens the understanding of heterogeneity between cells and cell states. For example, GraphSCC [225] employs graph convolutional networks and a denoising autoencoder network to capture high-order structural relations between cells. It uses a dual self-supervised module to get clusters of scRNA-seq data.

4.3. Link Prediction

Link prediction is one of the essential graph applications for biomedical graph analysis. The goal is to predict missing links that may appear in graphs (e.g., possible links between two drug molecules may appear in the future). In addition, link prediction can also help identify spurious links and understand the graph evolution mechanism.

4.3.1. Pharmaceutical Data Analysis

The design and manufacture of drugs are well known to be expensive and time-consuming. Therefore, researchers require more advanced computation technologies for more accurate predictions of interactions. To name a few, *drug-target interaction prediction studies*, *drug-disease interaction prediction studies*, and *drug-drug interaction prediction studies* can help researchers understand the mechanism of current drugs. Researchers can save drug development costs and increase productivity by exploring their possible unknown off-target activities and possible side effects of combining multiple drugs [251].

Drug-target interaction prediction. Drug-target interactions (DTIs) prediction, as a key task in genomic drug discovery, aims to discover unknown interactions between drugs and protein targets in human bodies. Yamanishi et al. [203] utilize an eigenvalue factorization algorithm to predict unknown DTIs from a bipartite graph, including the chemical structure and genomic sequence information. Other approaches [26, 54] develop matrix factorization-based models to predict potential DTIs. Cobanoglu et al. [26] use the probabilistic matrix factorization method for predicting potential drug-target interactions, which is particularly useful for analyzing large interaction graphs. Hao et al. [54] consider a logistic matrix factorization algorithm to predict potential drug-target interactions. Ezzat et al. [36] apply LE [4], SVD [47], and PLS [31] techniques to get embeddings of the DTIs graph. The final predictions are derived by further using the decision tree and kernel ridge regression. DDR [133] inputs different graph-based features extracted from the DTIs heterogeneous graph into a random forest model for more accurate drug target prediction.

Recently, to further improve the model capability of DTIs prediction, DTiGEMS+ [169] augments the known drug–target interactions graph with a drug–drug similarity graph and a target–target similarity graph, and uses node2vec [51] to learn representations of drugs and targets for drugs-target interaction prediction. To initiatively learn topology-preserving embeddings of drugs and targets for DTIs prediction, model NeoDTI [178] develops a new nonlinear end-to-end learning model that integrates diverse information from heterogeneous graph data. Similar to NeoDTI [178], (EEG)-DTI [140] uses graph convolutional networks to predict DTIs based on the low-dimensional feature representation of drugs and targets from a heterogeneous graph containing multiple types of biological entities. GraphDTA [127] uses graph neural networks to predict drug-target affinity, and Zhao et al. [241] utilize the GCN to learn latent representations of drug-protein pairs for drugs-target interaction prediction.

Drug-disease interaction prediction. Identifying potential drug-disease interactions is an integral part in the process of drug discovery. Investigation of complex relation-

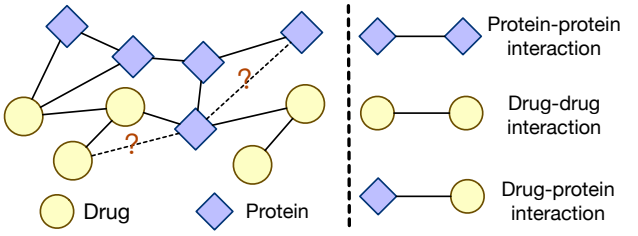

Figure 21: Link prediction. The goal of link prediction tasks in the biomedical field is to predict missing links (e.g., Drug-related tasks, Gene-related tasks, etc.). In addition, link prediction can also help identify spurious links in biomedical graphs and understand the graph evolution mechanism.

ships between drugs and diseases can find potential new target diseases for an existing drug and apply the newly identified drug to the treatment. CI-PMF [207] constructs three causal graphs for cardiovascular diseases, diabetes mellitus and neoplasms, respectively. It uses an inference-probabilistic matrix factorization method to make accurate drug-disease predictions. Dai et al. [29] integrate drug-gene interactions, disease-gene interactions, and gene-gene interactions to predict novel drug indications. SCMFDD [235] incorporates known drug–disease associations, drug features, and disease semantic information and proposes a similarity-constrained matrix factorization method to predict drug-disease associations. LAGCN [221] uses heterogeneous graph convolutions to learn the embeddings of drugs and diseases for the drug-disease association prediction.

Drug-drug interaction prediction. When drug combinations are used to treat complex or coexisting diseases, the pharmacological activity of one drug may change due to the interaction with other drugs, leading to unanticipated drug-drug interactions (DDIs). Furthermore, DDIs may lead to preventable Adverse Drug Reactions (ADRs) and cause a significant burden on the patients' health or increase the risks of morbidity. Zitnik and Zupan [256] work on multiplex drug data to predict connections between clinical manifestations of diseases and their underlying molecular signatures. Abdelaziz et al. [1] embed a drug knowledge graph using a large-scale logistic regression model to predict potential drug-drug interactions. Later, Decagon [254] proposes a deep autoencoder method to predict labeled drug-drug interactions. Inspired by Decagon, Nováček and Mohamed [132] follow the multi-phase procedure to predict polypharmacy side-effects. Recently, KGNN [105] proposes a spatial-based GNN method to aggregate topological neighborhood information from drugs' local receptive fields to get informative embeddings for predicting potential DDIs. Authors in [79] propose various knowledge graph-based embedding methods and classic machine learning classifiers for predicting drug–drug interactions. In addition, attention-based methods such as [114, 22] design attention mechanisms to automatically highlight the deterministic factors in biomedical knowledge learning, providing an attentive capability to estimate the interactions between drugs.

4.3.2. Multi-omics Data Analysis

With the advancement of sequencing technologies, researchers could obtain information on different levels of biomolecules. Therefore, biomedical research relies on data generated at different levels, which are referred to *multi-omics data*. Multi-omics data analysis can guide the discovery of potential associations among RNA, genes, and proteins and thus revolutionize the field of medicine and biology.

Gene-disease association prediction. Understanding the role of genes in human disease provides essential information for the prevention, diagnosis and treatment of diseases. The gene-disease association associations, coupled with the existing gene-related and disease-related databases, enable the prediction of unknown associations by computational approaches. To achieve this, Zhu et al. [248] construct a heterogeneous gene-disease association graph via integrating multiple biomedical knowledge bases. They define a novel cluster loss function and a dropout mechanism to improve the generalization ability. Wang et al. [186] considers six types of graph embedding methods to extract the information from the heterogeneous graphs with gene-disease associations, gene-chemical associations, and disease-chemical associations. GCN-MF [53] combines the GCN and matrix factorization and defines a margin control loss function that captures non-linear interactions and reduces the effect of sparsity to solve the gene-disease association prediction task. KGED [25] applies a biological knowledge graph with entity descriptions to infer gene-disease relationships.

RNA-disease association prediction. RNAs have a significant role in identifying various complex human diseases. Identifying the potential associations between RNAs and diseases is critical to reveal the mechanism of biological processes and complicated diseases to help disease diagnosis, therapy, prognosis, and monitoring [71]. However, the number of known disease-related RNAs is still very limited, and many experimental identifications are time-consuming and labor-intensive. Therefore, researchers have focused on developing useful methods to predict such associations. In the non-coding RNA family, microRNA (miRNA) is closely associated with a variety of human diseases. The methods in [71, 70] learn a heterogeneous biomedical graph to predict potential miRNA-disease associations using Random Forest (RF) classifier. Recently, Zhang et al. [231] present a novel multiple meta-paths fusion graph embedding model that extracts all meta-path instances connecting miRNAs with diseases to predict unidentified miRNA-disease associations.

Similarly, Liu et al. [107] combine the gate recurrent unit (GRU) model and the multi-layer perceptron (MLP) model, achieving advancing performance in the miRNA-disease association prediction. In addition, some other attempts fuse different effective techniques to predict miRNA-disease associations. For instance, Authors in [201] apply the Structural Deep Network Embedding [179] (SDNE) on the observed bipartite association graph for embedding learning; and CNNMDA [201] integrates the similarity information of miRNAs and diseases, miRNA-disease associations to predict disease miRNAs by applying the convolutional neural

network architecture. MMGCN [168] employs a GCN encoder to obtain the features of miRNA and disease in different similarity views and utilizes multi-channel attention, which adaptively learns the importance of different features, to enhance the learned latent representations for potential miRNA-disease associations prediction.

In many aspects, diseases' physiological and pathological processes are associated with Long non-coding RNA (lncRNA). To predict candidate disease-related lncRNAs, *Xun et al.*[199] adapt the autoencoder to learn the topological information within the heterogeneous lncRNA-disease-miRNA network; and the author [200] also fuses convolutional neural networks with the attention mechanisms to learn the embeddings of the lncRNA–disease pairs. Besides, to learn better embeddings for lncRNAs, model GCAN [228] operates the computation process from the global structure of disease-RNAs graph and *Zhang et al.*[229] use adversarial regularization to discover feature representation distribution of the latent nodes in disease-RNA graphs. Zhao et al. [238] adopt several graph embedding methods to enrich the informativeness of latent representations for lncRNAs and miRNAs.

Protein-protein interaction prediction. Based on the observed protein-protein interactions (PPIs) graphs, authors in [86] utilize the MDS [62] model to evaluate the confidence of existing PPIs and further predict new ones. You et al. [215] use the Isomap [5] model to embed the PPI graph into a low-dimensional metric space by preserving geodesic distances among protein nodes. PPI prediction is then made by measuring the similarity among proteins in the embedding space. Later work NCSE [249] adaptively learns Euclidean embeddings under the simple geometric assumption of PPI graphs to assess the reliability of PPIs. Recently, Zhong and Rajapakse [242] learn representations from the new knowledge sources, i.e., Gene Ontology Annotation (GOA) graphs, to estimate missing and spurious protein-protein interactions. Authors in [209] stack GCNs to embed a protein interaction graph into a vector space to remove less credible PPIs.

4.3.3. Protection and Prediction against Infectious Diseases

Outbreaks of infectious diseases have huge impacts on public health and social economy. For example, COVID-19 Pandemic has caused heavy human casualties and economic damage globally. In states with pandemics, the major challenge is tracking and tracing infected individuals and predicting the outbreaks' extent and duration. Accurate analysis and prediction of corresponding data improve the efficiency of decision-making, such as the reasonable allocation of limited medical resources.

Graph embedding methods they used prove the great potential for reasoning these related problems, such as contact tracking and disease prediction. Holme [63] construct a static graph for graph epidemiology from temporal contact data. SIGN [96] puts forward a GCN-based framework for the problem of source identification in a given graph with different infection rates. However, classic graph analysis

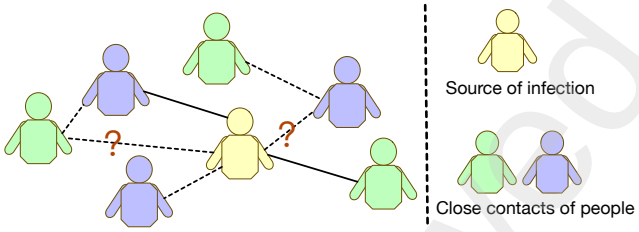

Figure 22: A representative work on dynamic link prediction in biomedicine is infectious disease prediction. The yellow icon represents the infectious source (patient) and the blue and green icons represent close contacts. Dynamic linkage prediction can be used to predict the probability of close contact suffering from an infectious disease.

methods do not consider contact data's temporal properties. Instead of neglecting temporal information modeling, Koher et al. [85] focus on the temporal graphs to predict disease outbreaks. *Kapoor et al.*[78] use GNN methodology for mobility data to predict COVID-19 cases. In addition, La Gatta et al. [89] jointly use the LSTM and GCN models to analyze the graph series. They predict the spread of infection and how different lock-down strategies influence epidemic diffusion.

4.4. Visualization

Biomedical graphs are usually multi-relational and complex, and graph visualization techniques play a crucial role in the mechanistic understanding of biomedical relations. In a similar vein as clustering, recent developments in graph embedding methods provide a new, potent paradigm for graph visualization problems. This allows researchers to understand relationships between nodes and visualize multi-relational node clusters.

For instance, in clinical studies, diseases caused by multiple etiologies are usually difficult to be diagnosed. Graph-based methods have become a powerful and popular approach for analyzing brain diseases, such as schizophrenia and mild traumatic brain injury [134]. In these methods, different brain regions are replaced by graph nodes and edges that denote the structural connectivity and functional connections in white matter. Li et al. [101] propose a graph encoder-decoder framework as well as a multi-objective loss function to jointly learn low-dimensional node representations, which integrates information from both nodes attributes and brain graph topology. Liu et al. [109] use graph convolutions to learn the structural and functional joint embeddings. It preserves the community structure of brain graphs and uses Siamese architecture to guide the learning process. Similar to the model SCP-GCN [109], researchers in [113] propose a Siamese GCN framework consisting of two twin graph convolutional networks to learn the brain graph representations. In addition, Hi-GCN [74] proposes a hierarchical GCN framework to learn the graph feature embedding of brain graphs by considering the graph topology information and biological associations. After getting node embeddings, researchers can combine low-dimensional representations with well-known techniques such as t-SNE or PCA to generate information visualizations. Graph visualization makes it eas-

ier for researchers to analyze the causal mechanisms of diseases, thus assisting physicians in decision-making.

The proper tools to visualize biomedical datasets can help researchers discover critical findings from complex information. Authors in [120, 157] visualize the citation dataset as a knowledge graph for COVID-19 analysis. It [120] develops a tool named *Covid Linked Data Visualizer* that encompasses a set of specialized visualization techniques, such as node-edge diagrams, cluster-based visualization, etc.

4.5. Graph Level Applications

The majority of studies have concentrated on extracting topological information from the neighborhoods of nodes and edges. However, kernel graphs can provide more features beyond a single node neighborhood, introducing additional information for downstream tasks.

4.5.1. Drug and Protein Structure Generation

The discovery of new molecular structures with desired properties has been a fundamental problem in drug discovery. However, the design and manufacture of the new molecular structures require not only the generation of chemically valid molecular structures but also the optimization of their chemical combinatorial complexity. As a result, domain experts are motivated to develop more advanced models for generating molecular structures, producing a promising way to accelerate the drug discovery process.

Early work such as JT-VAE [76] uses a VAE-based method to generate the junction tree-structured chemical substructures and then combine them into a molecule with desirable properties by a graph message passing network. Then authors in [151, 224, 112] come up with flow-based generative models for the molecular graph generation. Concretely, GraphAF [151] combines the advantages of both autoregressive and flow-based approaches for generating new molecular graphs. The method formulates graph generation as a sequential decision process, generating a new atom in each step and then determining the bonds between generated and existing atoms. MoFlow [224] proposes an invertible flow-based generative model which first generates bonds (edges) and atoms (nodes) through a Glow-based model and a novel graph conditional flow. Moreover, the model assembles bonds and atoms into a chemically valid molecular graph with a posthoc validity correction mechanism. And GF-VAE [112] equips VAE with a lightweight flow model as its decoder to accelerate the training process. Furthermore, GraphDF [111] proposes a novel discrete latent variable model for the molecular graph generation. The method uses invertible modulo shift transforms to map discrete latent variables to graph nodes and edges. MGM [117] develops a masked graph model for molecule generation, which learns a distribution over graphs by capturing conditional distributions over unobserved graph components. On the other hand, protein structure generation is another critical direction in the biomedical field. *Ingraham et al.*[68] modify the GAT model to efficiently capture the complex dependencies in proteins for rapid and targeted biomolecular design. ProteinSolver [159] phrases protein design challenge as a constraint satisfaction problem (CSP), akin to Sudoku puzzles, and presents a deep graph neural network-based method to precisely design sequences that fold into a predetermined shape.

4.5.2. Protein Properties Prediction

Martino et al. [119] use the theory of topological data analysis to extract a set of features from protein contact networks to predict proteins' physiological function. Model ProteinGCN [149] uses GCN to learn the atom embeddings from the protein structure graph. Then the method pools the atom embeddings to generate residue level embeddings and a global protein embedding to estimate the quality of protein models. GeomGCL [98] proposes a novel graph contrastive learning-based method utilizing the dual-view geometric message of the molecule to predict molecule function. In addition, DeepFRI [46] and GAT-GO [90] provide a graph convolutional network and a graph attention network method for protein function prediction by leveraging sequence features and structure information of protein.

4.5.3. Molecule Interaction Prediction

Recent advances in modeling molecular structural information have enabled accurate prediction of the properties of most molecules, such as drugs and proteins. However, the functions of molecules are intrinsically linked to their interactions. For example, protein-protein interactions play a vital role in the mechanistic understanding of cells. Therefore, determining the structure of protein complexes to understand the molecular mechanisms of interaction can provide key insights for understanding molecule properties. Fout et al. [39] learn effective latent representations that represent the three-dimensional structure of a protein of interest by performing convolution over a local node neighborhood and then consider the prediction of interfaces between proteins. MASIF [42] develops a geometric deep learning-based framework to capture important fingerprints for specific biomolecular interactions. PInet [28] develops a unified Geometric Deep Neural Network, which leverages the properties of both data- and physics-driven methods to improve performance in predicting protein interaction interfaces.

Furthermore, accurate prediction of target-ligand interactions is the most important step in the drug discovery process. Lim et al. [103] introduce a distance-aware graph attention algorithm to differentiate various types of inter-molecular interactions and extract the graph feature of inter-molecular interactions directly from the 3D structural information on the protein-ligand binding pose for accurate predictions of drug-target interaction. Graph-CNN [171] uses an unsupervised graph-autoencoder to learn fixed-size representations of protein pockets from a set of representative druggable protein binding sites and trains two Graph-CNNs to automatically extract features from pocket graphs and 2D ligand graphs for drug-target interactions prediction.

4.5.4. Biomedical Graph Classification

The classification of biomedical graphs modeling different tissue structures is of great importance for diagnostic

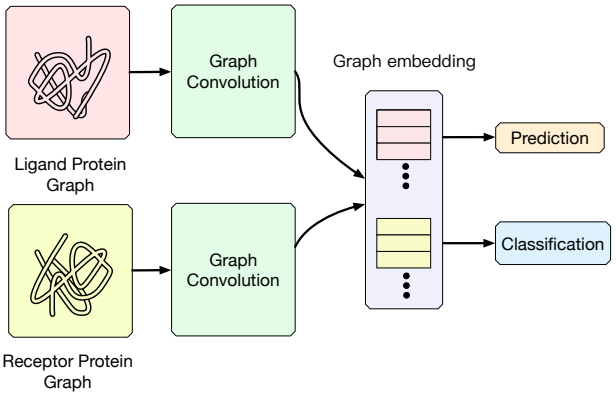

Figure 23: Graph-level applications utilize structural information extracted from biomolecules to provide additional features beyond the neighborhood of a single node, which is helpful for downstream tasks [39].

studies and disease prevention. For example, Studer et al. [160] use a GNN-based approach on cell-graphs to classify intestinal glands as normal or dysplastic, which may help prevent colorectal cancer. Netpro2vec [118] proposes a neural embedding framework based on probability distribution representations of graphs for biomedical graph classification. Zhou et al. [243] develop an interpretable GCN framework and utilize a modified Gradient Class Activation Mapping (Grad-CAM) technique for the identification and classification of Alzheimer's disease (AD). In addition, model MS-GWNN [232] leverages a novel graph convolutional neural network on histopathological images for breast cancer classification. It encodes the multi-scale contextual interactions in the whole pathological slide by aggregating features at different scales.

4.6. General Biomedical Representation Learning

Biomedical datasets are usually large, complex, and heterogeneous. The complexity and heterogeneity make traditional methods difficult to process intricate and interrelated information. Thus the basic task is to learn general biomedical embeddings, minimizing data redundancy and reducing the number of variables with minimal loss of information.

For example, SCRL [100] firstly constructs a so-called Cell-ContexGene graph based on the scRNA-seq data and a Gene-ContexGene graph based on pathway annotations, respectively. Then it uses a joint bipartite graph embedding method based on LINE [167] to learn low-dimensional representations for both the cells and the genes. Authors in [184] propose multi-modal methods using graph convolution networks to learn relation-aware representations for diagnosing chest CT images. Recent model PACER [183] focuses on the heterogeneous graph comprised of gene expression and drug response-gene information; and Zhu et al. [248] study the heterogeneous graph consisting of multiple biomedical knowledge bases. In addition, MoCL [163] and KCL [37] introduce contrastive learning-based methods, which utilize different views of knowledge to assist biomedical representation learning.

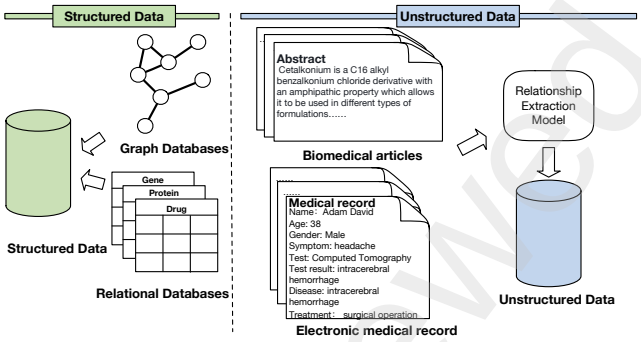

Figure 24: Structured data and unstructured data.

5. Open-Source Implementations and Datasets

5.1. Open-Source Implementations

We collect the open source implementations of representative important graph embedding methods in Table 5[1].

5.2. Biomedical Datasets

Biomedical data comes in various forms, such as vital signs, laboratory results, electronic medical records, and various clinical images [128]. Although humans can easily comprehend all these forms of data, structured data is generally more suitable for biological computation. Generally, we categorize the data into **structured data** and **unstructured data**. As shown in figure 24, structured data can be interpreted and stored in a fixed schema, making it amenable to rapid processing by programs without extensive human knowledge for data annotation. The most commonly used information, such as laboratory results, demographic information, and diagnosis codes, can be stored in structured tables. Conversely, unstructured data is challenging to integrate into a uniform data structure, as this necessitates significant human resources for data preprocessing. However, unstructured raw data usually contain more contextual information, such as biomedical publications and their mutual citations. Consequently, unstructured data is usually manually annotated for knowledge mining in accordance with specific downstream tasks. In this section, we summarize widely used structured and unstructured datasets in the biomedical domain.

5.2.1. Structured Relational Data

DRKG[2]: As a comprehensive biological knowledge graph, Drug Repurposing Knowledge Graph (DRKG) relates genes, compounds, diseases, biological processes, side effects and symptoms, including 97,238 entities belonging to 13 entity-types; and 5,874,261 triplets belonging to 107 edge-types.
Hetionet[3]: Hetionet is a heterogeneous graph of biomedical knowledge integrated from 29 public databases. It contains 47,031 nodes of 11 types and 2,250,297 edges of 24 types, representing compounds, diseases, genes, etc.

[1] The open source implementations are put in references.
[2] https://github.com/gnn4dr/DRKG
[3] https://het.io

Table 5
Open source implementations.

Graph Type	Catagory	Methods
Homogeneous Graph	Matrix factorization-based methods	GF [3], GraRep [13], TADW [205], HOPE [136], LE [4], MDS [62], LLE [147], IsoMAP [5], AgLPP [75]
	Random walk-based methods	DeepWalk [143], node2vec [51], TriDNR [138], DeepCas [94], HARP [16]
	Deep learning-based methods	SDNE [179], DNGR [14], GAE [82], VGAE [82], GC-MC [7], G2G [8], DVNE [245], ARVGAE [137], NetRAs [220], Spectral CNN [11], GCN [83], GraphSage [52], GAT [176], LGCN [43], FastGCN [18], StoGCN [21], ClusterGCN [23], DGI [177], GMI [142], MVGRL [55], GCC [144], GRACE [252], GCA [253], GraphCL [214], JOAO [213], DAGA [172], AutoGCL [211], AutoSSL [77], SimGRACE [195], AF-GRL [92]
Heterogeneous Graph	Proximity preserving methods	PTE [166], MVE [145], AspEM [152], HEER [153], HeGAN [65], Metapath2vec [34], HIN2vec [40], BHIN2vec [93], HHNE [189], MetaGraph2Vec [227], mg2vec [234]
	Relation learning methods	TransE [10], TransH [190], TransR [106], TransD [72], TranSparse [73], RESCAL [130], DistMult [204], HolE [129], ComplEx [175], SimplE [81], RotatE [164]
	Message passing methods	R-GCN [150], HAN [187], MAGNN [187], HetSANN [64], HGT [67], HetGNN [226], GATNE [15], NSHE [239], GTN [223], RSHN [250], HeCo [188]
Dynamic Graph	Matrix factorization-based methods	DHPE [246], Times [236]
	Random walk-based methods	tNodeEmbed [154], CTDNE [126], T-EDGE [104], HTNE [257]
	Deep learning-based methods	DynGEM [50], E-LSTM-D [20], NetWalk [219], dyngraph2vec [49], DyGGNN [165], EvolveGCN [139], GC-LSTM [19], JODIE [88], DySAT [148], Know-Evolve [173]

PubMed-diabetes[4]: PubMed-diabetes (denoted as PM-D) is a citation network consisting of 19,717 scientific publications regarding diabetes and 44,338 edges (citations).
PPI[5]: PPI is a dataset for protein-protein interactions, which contains 20 graphs for training, 2 graphs for testing, and 2 graphs for validation. The average number of nodes in each graph is 2,372.67, and each node has multiple labels from 121 classes. The average number of edges is 34,113.17.
MUTAG[6]: MUTAG dataset is a collection of nitroaromatic compounds with the aim of predicting their mutagenicity on Salmonella typhimurium.
NCI1, **NCI33**, **NCI83**, **NCI109** [6]: NCI1 is a knowledge base that contains 4,110 chemical compounds that are screened for activity against non-small cell lung cancer. NCI33, NCI83, NCI109 contain 2,843, 3,867, and 4,127 chemical compounds that are screened for activity against melanoma cancer, breast cancer cell lines, and ovarian cancer cell lines, respectively.
D&D, **PROTEIN**[7]: DD represents proteins in the graph where labels are enzymes. PROTEIN represents proteins in the graph where labels are non-enzymes.
PTC [170]: PTC dataset contains 344 chemical compounds in which classes indicate the carcinogenicity on rats.
ENZYMES [196]: ENZYMES is a dataset containing 600 protein tertiary structures obtained from the BRENDA enzyme database.
Infectious[8]: Infectious is a dynamic graph dataset recording the interaction of visitors in the exhibition. When there are at least 20 seconds of face-to-face communication between visitors, the edges of the graph will be established and have an accurate time stamp.
DrugBank[9]: DrugBank is a comprehensive database that combines detailed drug data with drug target information. It contains 14,460 drug, 18,906 drug-target associations and 5,301 drug-protein associations.
IntAct[10]: IntAct is a database for storing, presenting and analyzing protein interactions. It contains around 2,200 binary and complex interactions imported from the literature.
DGIdb[11]: The Drug-Gene Interaction Database (DGIdb) is a database of drug-gene interactions. It contains over 40,000 genes and 10,000 drugs from publications, databases, and other web-based sources.
STRING[12]: STRING is a database of protein-protein interactions. These interactions include both direct (physical) and indirect (functional) associations. The database currently covers 24,584,628 proteins from 5,090 organisms.

5.2.2. Unstructured Data.

MEDLINE[13]: MEDLINE, the bibliographic database of National Library of Medicine (NLM), offers over 27 million references in journal articles in the biomedical field.
PubMed[14]: PubMed is a citation database which contains more than 26 million MEDLINE scholarly work, such as online books and journals about life science.
MIMIC-III[15]: MIMIC-III is a de-identified EHR database

[4] https://linqs.soe.ucsc.edu/data
[5] http://snap.stanford.edu/graphsage/ppi.zip
[6] https://ls11-www.cs.uni-dortmund.de/people/morris/graphkerneldatasets
[7] https://chrsmrrs.github.io/datasets/docs/datasets/
[8] http://konect.uni-koblenz.de/networks/sociopatterns-infectious
[9] https://go.drugbank.com/
[10] https://www.ebi.ac.uk/intact/
[11] https://www.dgidb.org/
[12] https://www.string-db.org/
[13] https://www.nlm.nih.gov/medline/index.html
[14] https://pubmed.ncbi.nlm.nih.gov/
[15] https://mimic.mit.edu/

Table 6
Dataset Statistics.

Dataset	Nodes	Edges	Classes	Features	Graph Type
DRKG	97,238	5,874,261	-	-	Knowledge Graph
Hetionet	47,031	2,250,297	-	-	Knowledge Graph
PM-D	19,717	44,338	3	500	Citation Graph
PPI	56,944	818,716	121	50	Bio-chemical Graph
MUTAG	17	19	2	7	Bio-chemical Graph
NCI1	29	32	2	37	Bio-chemical Graph
NCI109	29	-	-	28	Bio-chemical Graph
NCI33	30	-	-	29	Bio-chemical Graph
NCI83	29	-	-	28	Bio-chemical Graph
DD	284	715	2	82	Bio-chemical Graph
PROTEIN	39	72	2	4	Bio-chemical Graph
PTC	25	-	2	19	Bio-chemical Graph
ENZYMES	32	63	6	6	Biological Graph

that includes more than 40,000 patients. The database consists of information on demographics, bedside vital sign measurements, laboratory test results, etc.

6. Challenges and Future Directions

Although current graph embedding methods have demonstrated effectiveness, there remain challenges and potential areas for exploration in the context of biomedical tasks.

Data sparsity and low-quality. (1) The privacy and specialization of biomedical data poses a significant challenge in obtaining processed data, e.g., annotated labels, for practical clinical applications. For example, some diseases (e.g., moyamoya disease) only affect specific groups of people in real clinical scenarios, so the number of patients is very limited. Additionally, strict privacy policies restrict access to the real clinical data, causing to incomplete and sparse data issues. Moreover, the biomedical field often involves unstructured and complex data that require specialized domain expertise for processing, which differs from general application domains. However, manual labeling is usually both expensive and time-consuming, leading to massive raw data that requires processing for downstream realistic clinical applications.

(2) The quality of input data significantly affects the outcome of graph embedding methods. However, biomedical data are usually are generally noisy, complex, and uneven. For instance, in the context of gene analysis, the number of normal genes is usually much larger than that of disease-causing genes, which are of primary interest for research. Using imbalanced data for pharmaceutical and multi-omics data analysis may cause unsatisfactory outcomes. Moreover, biomedical data encompass multiple data modalities, including digital signals, clinical histories, medical images, etc. Therefore, high-quality biomedical data can simplify the complexity of learning biomedical graph embeddings for downstream tasks.

Model scalability. Although existing models generally have presented good performance, the model scalability is still a major challenge to deal with. Scalability can be divided into *data volume scalability* and *dataset scalability* [123]. *Data volume scalability* involves scaling the graph to millions or billions of nodes for model learning. Despite the effectiveness of existing graph embedding methods, deploying them to large-scale data still presents challenges.

Dataset scalability, on the other hand, refers to the use of different forms of datasets in most previous studies. For example, since medical record management systems of various hospitals are almost independent of each other, the effect of the same model on different datasets may vary greatly. Thus, researchers may need to consider improving model scalability for different datasets.

Model credibility and interpretability. In the field of biomedicine, the credibility and interpretability of models are other important challenges. Specifically, models are expected to be both accurate and interpretable to serve as a medical reference for clinicians in making decisions related to medical treatment, hospitalization, clinical surgery, and other aspects of healthcare. In realistic clinical scenarios, interpretability is an important feature for clinical decision-making.

Robustness. Most traditional graph embedding models are practically vulnerable to defense the adversarial perturbations or attacks, making these algorithms unable to produce stable representations and results. Therefore, the robustness of graph learning models has become a significant topic to work on. The model robustness requires attention for realistic deployment.

7. Conclusion

Graph embedding methods aim to learn compact and informative representations for graph analysis, offering a powerful solution for traditional graph-based machine learning problems. The growth of relational data in the biomedical field has accelerated in recent years, leading to increased attention on applying graph embedding techniques in biomedical research. However, despite the heightened interest in this area, the full potential of graph embedding for biomedical graph analysis has yet to be fully explored. In this paper, we present a comprehensive review of the latest trends and developments in the use of graph data in the biomedical field, focusing on the advances in graph embedding techniques and their application to various biomedical tasks. We also identify promising avenues for future research to advance the field and ultimately improve human health care.

References

[1] Abdelaziz, I., Fokoue, A., Hassanzadeh, O., Zhang, P., Sadoghi, M., 2017. Large-scale structural and textual similarity-based mining of knowledge graph to predict drug–drug interactions. J WEB SEMANT 44, 104–117.

[2] Abu-El-Haija, S., Perozzi, B., Al-Rfou, R., Alemi, A.A., 2018. Watch your step: Learning node embeddings via graph attention, in: NeurIPS, pp. 9198–9208. URL: https://github.com/benedekrozemberczki/attentionwalk.

[3] Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., Smola, A.J., 2013. Distributed large-scale natural graph factorization, in: WWW. URL: https://github.com/thunlp/OpenNE.

[4] Anderson Jr, W.N., Morley, T.D., 1985. Eigenvalues of the laplacian of a graph. Linear and multilinear algebra 18, 141–145. URL: https://github.com/palash1992/GEM.

[5] Balasubramanian, M., Schwartz, E.L., Tenenbaum, J.B., de Silva, V., Langford, J.C., 2002. The isomap algorithm and topological stability. Science 295, 7–7. URL: https://github.com/drewwilimitis/Manifold-Learning.

[6] Barros, C.D., Mendonça, M.R., Vieira, A.B., Ziviani, A., 2021. A survey on embedding dynamic graphs. CSUR 55, 1–37.

[7] Berg, R.v.d., Kipf, T.N., Welling, M., 2017. Graph convolutional matrix completion. arXiv:1706.02263 URL: https://github.com/riannevdberg/gc-mc.

[8] Bojchevski, A., Günnemann, S., 2018. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking, in: ICLR. URL: https://github.com/abojchevski/graph2gauss.

[9] Bordes, A., Glorot, X., Weston, J., Bengio, Y., 2014. A semantic matching energy function for learning with multi-relational data. Machine Learning 94, 233–259.

[10] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O., 2013. Translating embeddings for modeling multi-relational data, in: NeurIPS, pp. 2787–2795. URL: https://github.com/thunlp/OpenKE.

[11] Bruna, J., Zaremba, W., Szlam, A., LeCun, Y., 2014. Spectral networks and locally connected networks on graphs, in: ICLR. URL: https://github.com/mdeff/cnn_graph.

[12] Cai, H., Zheng, V.W., Chang, K.C.C., 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. TKDE 30, 1616–1637.

[13] Cao, S., Lu, W., Xu, Q., 2015. Grarep: Learning graph representations with global structural information, in: CIKM, pp. 891–900. URL: https://github.com/benedekrozemberczki/GraRep.

[14] Cao, S., Lu, W., Xu, Q., 2016. Deep neural networks for learning graph representations, in: AAAI. URL: https://github.com/ShelsonCao/DNGR.

[15] Cen, Y., Zou, X., Zhang, J., Yang, H., Zhou, J., Tang, J., 2019. Representation learning for attributed multiplex heterogeneous network, in: SIGKDD, pp. 1358–1368. URL: https://github.com/THUDM/GATNE.

[16] Chen, H., Perozzi, B., Hu, Y., Skiena, S., 2018a. Harp: Hierarchical representation learning for networks, in: AAAI. URL: https://github.com/GTmac/HARP.

[17] Chen, H., Yin, H., Wang, W., Wang, H., Nguyen, Q.V.H., Li, X., 2018b. Pme: projected metric embedding on heterogeneous networks for link prediction, in: SIGKDD, pp. 1177–1186.

[18] Chen, J., Ma, T., Xiao, C., 2018c. Fastgcn: Fast learning with graph convolutional networks via importance sampling, in: ICLR, OpenReview.net. URL: https://github.com/matenure/FastGCN.

[19] Chen, J., Wang, X., Xu, X., 2022. Gc-lstm: Graph convolution embedded lstm for dynamic network link prediction. Applied Intelligence 52, 7513–7528. URL: https://github.com/benedekrozemberczki/pytorch_geometric_temporal.

[20] Chen, J., Zhang, J., Xu, X., Fu, C., Zhang, D., Zhang, Q., Xuan, Q., 2019a. E-lstm-d: A deep learning framework for dynamic network link prediction. SMC URL: https://github.com/jianz94/e-lstm-d.

[21] Chen, J., Zhu, J., Song, L., 2018d. Stochastic training of graph convolutional networks with variance reduction, in: ICML, pp. 941–949. URL: https://github.com/thu-ml/stochastic_gcn.

[22] Chen, X., Liu, X., Wu, J., 2019b. Drug-drug interaction prediction with graph representation learning, in: BIBM, pp. 354–361.

[23] Chiang, W.L., Liu, X., Si, S., Li, Y., Bengio, S., Hsieh, C.J., 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks, in: SIGKDD, pp. 257–266. URL: https://github.com/benedekrozemberczki/ClusterGCN.

[24] Choi, E., Bahadori, M.T., Song, L., Stewart, W.F., Sun, J., 2017. Gram: graph-based attention model for healthcare representation learning, in: SIGKDD, pp. 787–795.

[25] Choi, W., Lee, H., 2021. Identifying disease-gene associations using a convolutional neural network-based model by embedding a biological knowledge graph with entity descriptions. Plos one 16.

[26] Cobanoglu, M.C., Liu, C., Hu, F., Oltvai, Z.N., Bahar, I., 2013. Predicting drug–target interactions using probabilistic matrix factorization. Jnl of chemical information and modeling 53, 3399–3409.

[27] Cui, P., Wang, X., Pei, J., Zhu, W., 2018. A survey on network embedding. TKDE 31, 833–852.

[28] Dai, B., Bailey-Kellogg, C., 2021. Protein interaction interface region prediction by geometric deep learning. Bioinformatics 37, 2580–2588.

[29] Dai, W., Liu, X., Gao, Y., Chen, L., Song, J., Chen, D., Gao, K., Jiang, Y., Yang, Y., Chen, J., et al., 2015. Matrix factorization-based prediction of novel drug indications by integrating genomic space. COMPUT MATH METHOD M 2015.

[30] Dai, Y., Wang, S., Xiong, N.N., Guo, W., 2020. A survey on knowledge graph embedding: Approaches, applications and benchmarks. Electronics 9, 750.

[31] De Jong, S., 1993. Simpls: an alternative approach to partial least squares regression. Chemometr Intell Lab Syst 18, 251–263.

[32] De Winter, S., Decuypere, T., Mitrović, S., Baesens, B., De Weerdt, J., 2018. Combining temporal aspects of dynamic networks with node2vec for a more efficient dynamic link prediction, in: ASONAM, pp. 1234–1241.

[33] Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Cnns on graphs with fast localized spectral filtering, in: NeurIPS, pp. 3837–3845. URL: https://github.com/mdeff/cnn_graph.

[34] Dong, Y., Chawla, N.V., Swami, A., 2017. metapath2vec: Scalable representation learning for heterogeneous networks, in: SIGKDD, pp. 135–144. URL: https://github.com/apple2373/metapath2vec.

[35] Du, L., Wang, Y., Song, G., Lu, Z., Wang, J., 2018. Dynamic network embedding: An extended approach for skip-gram based network embedding., in: IJCAI, pp. 2086–2092.

[36] Ezzat, A., Wu, M., Li, X.L., Kwoh, C.K., 2017. Drug-target interaction prediction using ensemble learning and dimensionality reduction. Methods 129, 81–88.

[37] Fang, Y., Zhang, Q., Yang, H., Zhuang, X., Deng, S., Zhang, W., Qin, M., Chen, Z., Fan, X., Chen, H., 2022. Molecular contrastive learning with chemical element knowledge graph, in: AAAI, pp. 3968–3976.

[38] Ferreira, C.H.G., Ferreira, F.M., de Sousa Matos, B., de Almeida, J.M., 2019. Modeling dynamic ideological behavior in political networks. The Jnl of Web Science 7.

[39] Fout, A., Byrd, J., Shariat, B., Ben-Hur, A., 2017. Protein interface prediction using graph convolutional networks. NeurIPS 30.

[40] Fu, T.y., Lee, W.C., Lei, Z., 2017. Explore meta-paths in heterogeneous information networks for representation learning, in: CIKM, pp. 1797–1806. URL: https://github.com/csiesheep/hin2vec.

[41] Fu, X., Zhang, J., Meng, Z., King, I., 2020. Magnn: metapath aggregated graph neural network for heterogeneous graph embedding, in: WWW, pp. 2331–2341. URL: https://github.com/cynricfu/MAGNN.

[42] Gainza, P., Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M., Correia, B., 2020. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. NAT METHODS 17, 184–192.

[43] Gao, H., Wang, Z., Ji, S., 2018. Large-scale learnable graph convolutional networks, in: SIGKDD, pp. 1416–1424. URL: https://github.com/divelab/lgcn.

[44] Ge, W., Guo, W., Cui, L., Li, H., Liu, L., 2020. Detection of wrong disease information using knowledge-based embedding and attention, in: DASFAA, pp. 459–473.

[45] Gligorijević, V., Barot, M., Bonneau, R., 2018. deepnf: deep network fusion for protein function prediction. Bioinformatics 34.

[46] Gligorijević, V., Renfrew, P.D., Kosciolek, T., Leman, J.K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B.C., Fisk, I.M., Vlamakis, H., et al., 2021. Structure-based protein function prediction using graph convolutional networks. Nature communications 12, 1–14.

[47] Golub, G.H., Reinsch, C., 1971. Singular value decomposition and least squares solutions, in: Linear algebra. Springer, pp. 134–151.

[48] Gong, F., Wang, M., Wang, H., Wang, S., Liu, M., 2021. SMR: medical knowledge graph embedding for safe medicine recommendation. Big Data Res. 23, 100174.

[49] Goyal, P., Chhetri, S.R., Canedo, A., 2020. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. KBS 187. URL: https://github.com/palash1992/DynamicGEM.

[50] Goyal, P., Kamra, N., He, X., Liu, Y., 2018. Dyngem: Deep embedding method for dynamic graphs. arXiv:1805.11273 URL: https://github.com/paulpjoby/DynGEM.

[51] Grover, A., Leskovec, J., 2016. node2vec: Scalable feature learning for networks, in: SIGKDD, pp. 855–864. URL: https://github.com/eliorc/node2vec.

[52] Hamilton, W.L., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs, in: NeurIPS, pp. 1024–1034. URL: https://github.com/williamleif/GraphSAGE.

[53] Han, P., Yang, P., Zhao, P., Shang, S., Liu, Y., Zhou, J., Gao, X., Kalnis, P., 2019. Gcn-mf: disease-gene association identification by graph convolutional networks and matrix factorization, in: SIGKDD, pp. 705–713.

[54] Hao, M., Bryant, S.H., Wang, Y., 2017. Predicting drug-target interactions by dual-network integrated logistic matrix factorization. Sci. Rep. 7, 1–11.

[55] Hassani, K., Khasahmadi, A.H., 2020. Contrastive multi-view representation learning on graphs, in: ICML, pp. 4116–4126. URL: https://github.com/kavehhassani/mvgrl.

[56] Hawkes, A.G., 1971. Spectra of some self-exciting and mutually exciting point processes. Biometrika 58, 83–90.

[57] He, X., Niyogi, P., 2004. Locality preserving projections. NeurIPS 16, 153–160.

[58] He, Y., Song, Y., Li, J., Ji, C., Peng, J., Peng, H., 2019. Hetespaceywalk: A heterogeneous spacey random walk for heterogeneous information network embedding, in: CIKM, pp. 639–648.

[59] Henaff, M., Bruna, J., LeCun, Y., 2015. Deep convolutional networks on graph-structured data. arXiv:1506.05163 .

[60] Hitchcock, F.L., 1927. The expression of a tensor or a polyadic as a sum of products. J MATH PHYS 6, 164–189.

[61] Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y., 2019. Learning deep representations by mutual information estimation and maximization, in: ICLR.

[62] Hofmann, T., Buhmann, J., 1995. Multidimensional scaling and data clustering. NeurIPS , 459–466URL: https://github.com/drewwilimitis/Manifold-Learning.

[63] Holme, P., 2013. Epidemiologically optimal static networks from temporal network data. PLOS COMPUT BIOL 9, e1003142.

[64] Hong, H., Guo, H., Lin, Y., Yang, X., Li, Z., Ye, J., 2020. An attention-based gnn for heterogeneous structural learning, in: AAAI, pp. 4132–4139. URL: https://github.com/didi/hetsann.

[65] Hu, B., Fang, Y., Shi, C., 2019. Adversarial learning on heterogeneous information networks, in: SIGKDD, pp. 120–129. URL: https://github.com/librahu/HeGAN.

[66] Hu, L., Wang, X., Huang, Y.A., Hu, P., You, Z.H., 2021. A survey on computational models for predicting protein–protein interactions. Briefings in bioinformatics 22, bbab036.

[67] Hu, Z., Dong, Y., Wang, K., Sun, Y., 2020. Heterogeneous graph transformer, in: WWW, pp. 2704–2710. URL: https://github.com/acbull/pyHGT.

[68] Ingraham, J., Garg, V., Barzilay, R., Jaakkola, T., 2019. Generative models for graph-based protein design. NeurIPS 32.

[69] Ivanov, S., Burnaev, E., 2018. Anonymous walk embeddings, in: ICML, PMLR. pp. 2191–2200.

[70] Ji, B.Y., You, Z.H., Chen, Z.H., Wong, L., Yi, H.C., 2020a. Nempd: a network embedding-based method for predicting mirna-disease associations by preserving behavior and attribute information. BMC Bioinform. 21, 1–17.

[71] Ji, B.Y., You, Z.H., Cheng, L., Zhou, J.R., Alghazzawi, D., Li, L.P., 2020b. Predicting mirna-disease association from heterogeneous information network with grarep embedding model. Sci. Rep. 10, 1–12.

[72] Ji, G., He, S., Xu, L., Liu, K., Zhao, J., 2015. Knowledge graph embedding via dynamic mapping matrix, in: ACL, pp. 687–696. URL: https://github.com/thunlp/OpenKE.

[73] Ji, G., Liu, K., He, S., Zhao, J., 2016. Knowledge graph completion with adaptive sparse transfer matrix, in: AAAI. URL: https://github.com/FrankWork/transparse.

[74] Jiang, H., Cao, P., Xu, M., Yang, J., Zaiane, O., 2020. Hi-gcn: A hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction. CBM 127, 104096.

[75] Jiang, R., Fu, W., Wen, L., Hao, S., Hong, R., 2016. Dimensionality reduction on anchorgraph with an efficient locality preserving projection. Neurocomputing 187, 109–118. URL: https://github.com/fuweijie/AGLPP.

[76] Jin, W., Barzilay, R., Jaakkola, T., 2018. Junction tree variational autoencoder for molecular graph generation, in: ICML, PMLR. pp. 2323–2332.

[77] Jin, W., Liu, X., Zhao, X., Ma, Y., Shah, N., Tang, J., 2022. Automated self-supervised learning for graphs, in: ICLR. URL: https://github.com/ChandlerBang/AutoSSL.

[78] Kapoor, A., Ben, X., Liu, L., Perozzi, B., Barnes, M., Blais, M., O'Banion, S., 2020. Examining covid-19 forecasting using spatio-temporal graph neural networks. arXiv:2007.03113 .

[79] Karim, M.R., Cochez, M., Jares, J.B., Uddin, M., Beyan, O., Decker, S., 2019. Drug-drug interaction prediction based on knowledge graph embeddings and convolutional-lstm network, in: BCB, pp. 113–123.

[80] Kazemi, S.M., Poole, D., 2018a. Simple embedding for link prediction in knowledge graphs, in: NeurIPS, pp. 4289–4300.

[81] Kazemi, S.M., Poole, D., 2018b. Simple embedding for link prediction in knowledge graphs, in: NeurIPS, pp. 4289–4300.

[82] Kipf, T.N., Welling, M., 2016. Variational graph auto-encoders. arXiv:1611.07308 URL: https://github.com/tkipf/gae, https://github.com/DaehanKim/vgae.

[83] Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks, in: ICLR. URL: https://github.com/thunlp/OpenNE.

[84] Klicpera, J., Weißenberger, S., Günnemann, S., 2019. Diffusion improves graph learning, in: NeurIPS, pp. 13333–13345.

[85] Koher, A., Lentz, H.H., Hövel, P., Sokolov, I.M., 2016. Infections on temporal networks—a matrix-based approach. PloS one 11, e0151209.

[86] Kuchaiev, O., Rašajski, M., Higham, D.J., Pržulj, N., 2009. Geometric de-noising of protein-protein interaction networks. PLOS COMPUT BIOL 5, e1000454.

[87] Kullback, S., Leibler, R.A., 1951. On information and sufficiency. The annals of mathematical statistics 22, 79–86.

[88] Kumar, S., Zhang, X., Leskovec, J., 2019. Predicting dynamic embedding trajectory in temporal interaction networks, in: SIGKDD, pp. 1269–1278. URL: http://snap.stanford.edu/jodie.

[89] La Gatta, V., Moscato, V., Postiglione, M., Sperli, G., 2020. An epidemiological neural network exploiting dynamic graph structured data applied to the covid-19 outbreak. TBD 7, 45–55.

[90] Lai, B., Xu, J., 2022. Accurate protein function prediction via graph attention networks with predicted structure information. Briefings in bioinformatics 23, bbab502.

[91] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

[92] Lee, N., Lee, J., Park, C., 2022. Augmentation-free self-supervised learning on graphs, in: AAAI, pp. 7372–7380. URL: https://github.com/namkyeong/afgrl.

[93] Lee, S., Park, C., Yu, H., 2019. Bhin2vec: Balancing the type of relation in heterogeneous information network, in: CIKM, pp. 619–628. URL: https://github.com/sh0416/BHIN2VEC.

[94] Li, C., Ma, J., Guo, X., Mei, Q., 2017a. Deepcas: An end-to-end predictor of information cascades, in: WWW, pp. 577–586. URL: https://github.com/chengli-um/DeepCas.

[95] Li, J., Dani, H., Hu, X., Tang, J., Chang, Y., Liu, H., 2017b. Attributed network embedding for learning in a dynamic environment, in: CIKM, pp. 387–396.

[96] Li, L., Zhou, J., Jiang, Y., Huang, B., 2021. Propagation source identification of infectious diseases with graph convolutional networks.

JBI 116, 103720.

[97] Li, M.M., Huang, K., Zitnik, M., 2022a. Graph representation learning in biomedicine and healthcare. Nature Biomedical Engineering , 1–17.

[98] Li, S., Zhou, J., Xu, T., Dou, D., Xiong, H., 2022b. Geomgcl: geometric graph contrastive learning for molecular property prediction, in: AAAI, pp. 4541–4549.

[99] Li, W., Zhang, H., Li, M., Han, M., Yin, Y., 2022c. Mgegfp: a multi-view graph embedding method for gene function prediction based on adaptive estimation with gcn. Briefings in bioinformatics 23.

[100] Li, X., Chen, W., Chen, Y., Zhang, X., Gu, J., Zhang, M.Q., 2017c. Network embedding-based representation learning for single cell rna-seq data. Nucleic acids research 45, e166–e166.

[101] Li, Y., Shafipour, R., Mateos, G., Zhang, Z., 2020. Supervised graph representation learning for modeling the relationship between structural and functional brain connectivity, in: ICASSP, pp. 9065–9069.

[102] Li, Y., Vinyals, O., Dyer, C., Pascanu, R., Battaglia, P., 2018. Learning deep generative models of graphs. arXiv:1803.03324 .

[103] Lim, J., Ryu, S., Park, K., Choe, Y.J., Ham, J., Kim, W.Y., 2019. Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation. J CHEM INF MODEL 59, 3981–3988.

[104] Lin, D., Wu, J., Yuan, Q., Zheng, Z., 2020a. T-edge: Temporal weighted multidigraph embedding for ethereum transaction network analysis. Frontiers in Physics 8, 204. URL: https://github.com/lindan113/T-EDGE.

[105] Lin, X., Quan, Z., Wang, Z.J., Ma, T., Zeng, X., 2020b. Kgnn: Knowledge graph neural network for drug-drug interaction prediction., in: IJCAI, pp. 2739–2745.

[106] Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X., 2015. Learning entity and relation embeddings for knowledge graph completion, in: AAAI. URL: https://github.com/thunlp/OpenKE.

[107] Liu, B., Zhu, X., Zhang, L., Liang, Z., Li, Z., 2021. Combined embedding model for mirna-disease association prediction. BMC Bioinform. 22, 1–22.

[108] Liu, H., Wu, Y., Yang, Y., 2017. Analogical inference for multi-relational embeddings, in: ICML, pp. 2168–2178.

[109] Liu, J., Ma, G., Jiang, F., Lu, C.T., Philip, S.Y., Ragin, A.B., 2019. Community-preserving graph convolutions for structural and functional joint embedding of brain networks, in: IEEE BigData, pp. 1163–1168.

[110] Liu, Z., Li, X., Peng, H., He, L., Philip, S.Y., 2020. Heterogeneous similarity gnn on electronic health records, in: IEEE BigData, pp. 1196–1205.

[111] Luo, Y., Yan, K., Ji, S., 2021. Graphdf: A discrete flow model for molecular graph generation, in: ICML, PMLR. pp. 7192–7203.

[112] Ma, C., Zhang, X., 2021. Gf-vae: A flow-based variational autoencoder for molecule generation, in: CIKM, pp. 1181–1190.

[113] Ma, G., Ahmed, N.K., Willke, T.L., Sengupta, D., Cole, M.W., Turk-Browne, N.B., Yu, P.S., 2019. Deep graph similarity learning for brain data analysis, in: CIKM, pp. 2743–2751.

[114] Ma, T., Xiao, C., Zhou, J., Wang, F., 2018. Drug similarity integration through attentive multi-view graph auto-encoders, in: IJCAI, pp. 3477–3483.

[115] Ma, Y., Guo, Z., Ren, Z., Tang, J., Yin, D., 2020. Streaming graph neural networks, in: SIGIR, pp. 719–728. URL: https://github.com/alge24/DyGNN.

[116] Mahdavi, S., Khoshraftar, S., An, A., 2018. dynnode2vec: Scalable dynamic network embedding, in: IEEE BigData, pp. 3762–3765.

[117] Mahmood, O., Mansimov, E., Bonneau, R., Cho, K., 2021. Masked graph modeling for molecule generation. Nat. Commun. 12, 1–12.

[118] Manipur, I., Manzo, M., Granata, I., Giordano, M., Maddalena, L., Guarracino, M.R., 2021. Netpro2vec: a graph embedding framework for biomedical applications. IEEE ACM T COMPUT BI 19, 729–740.

[119] Martino, A., Rizzi, A., Mascioli, F.M.F., 2018. Supervised approaches for protein function prediction by topological data analysis, in: IJCNN, IEEE. pp. 1–8.

[120] Michel, F., Gandon, F., Ah-Kane, V., Bobasheva, A., Cabrio, E., Corby, O., Gazzotti, R., Giboin, A., Marro, S., Mayer, T., et al., 2020. Covid-on-the-web: Knowledge graph and services to advance covid-19 research, in: SEMWEB, pp. 294–310.

[121] Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013a. Efficient estimation of word representations in vector space, in: ICLR.

[122] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013b. Distributed representations of words and phrases and their compositionality, in: NeurIPS, pp. 3111–3119.

[123] Mohamed, S.K., Nounu, A., Nováček, V., 2021. Biological applications of knowledge graph embedding models. Briefings in bioinformatics 22, 1679–1693.

[124] Narayan, A., Roe, P.H., 2018. Learning graph dynamics using deep neural networks. IFAC-PapersOnLine 51, 433–438.

[125] Nelson, W., Zitnik, M., Wang, B., Leskovec, J., Goldenberg, A., Sharan, R., 2019. To embed or not: network embedding as a paradigm in computational biology. Front. Genet. 10, 381.

[126] Nguyen, G.H., Lee, J.B., Rossi, R.A., Ahmed, N.K., Koh, E., Kim, S., 2018. Continuous-time dynamic network embeddings, in: WWW, pp. 969–976. URL: https://github.com/Shubhranshu-Shekhar/ctdne.

[127] Nguyen, T., Le, H., Quinn, T.P., Nguyen, T., Le, T.D., Venkatesh, S., 2021. Graphdta: Predicting drug–target binding affinity with graph neural networks. Bioinformatics 37, 1140–1147.

[128] Nicholson, D.N., Greene, C.S., 2020. Constructing knowledge graphs and their biomedical applications. COMPUT STRUCT BIOTEC 18, 1414–1428.

[129] Nickel, M., Rosasco, L., Poggio, T., 2016. Holographic embeddings of knowledge graphs, in: AAAI. URL: https://github.com/thunlp/OpenKE.

[130] Nickel, M., Tresp, V., Kriegel, H.P., 2011. A three-way model for collective learning on multi-relational data, in: ICML. URL: https://github.com/thunlp/OpenKE.

[131] Niepert, M., Ahmed, M., Kutzkov, K., 2016. Learning convolutional neural networks for graphs, in: International conference on machine learning, pp. 2014–2023.

[132] Nováček, V., Mohamed, S.K., 2020. Predicting polypharmacy side-effects using knowledge graph embeddings. AMIA Jt Summits Transl Sci Proc 2020, 449.

[133] Olayan, R.S., Ashoor, H., Bajic, V.B., 2018. Ddr: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. Bioinformatics 34, 1164–1173.

[134] Olejarczyk, E., Jernajczyk, W., 2017. Graph-based analysis of brain connectivity in schizophrenia. PLoS One 12, e0188629.

[135] Oniani, D., Jiang, G., Liu, H., Shen, F., 2020. Constructing co-occurrence network embeddings to assist association extraction for covid-19 and other coronavirus infectious diseases. JAMIA 27, 1259–1267.

[136] Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W., 2016. Asymmetric transitivity preserving graph embedding, in: SIGKDD, pp. 1105–1114. URL: https://github.com/ZW-ZHANG/HOPE.

[137] Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., Zhang, C., 2018. Adversarially regularized graph autoencoder for graph embedding, in: IJCAI, ijcai.org. pp. 2609–2615. URL: https://github.com/GRAND-Lab/ARGA.

[138] Pan, S., Wu, J., Zhu, X., Zhang, C., Wang, Y., 2016. Tri-party deep network representation. Network 11, 12. URL: https://github.com/shiruipan/TriDNR.

[139] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., Leiserson, C., 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs, in: AAAI, pp. 5363–5370. URL: https://github.com/IBM/EvolveGCN.

[140] Peng, J., Wang, Y., Guan, J., Li, J., Han, R., Hao, J., Wei, Z., Shang, X., 2021a. An end-to-end heterogeneous graph representation learning-based framework for drug–target interaction prediction. Briefings in bioinformatics .

[141] Peng, J., Xue, H., Wei, Z., Tuncali, I., Hao, J., Shang, X., 2021b. Integrating multi-network topology for gene function prediction using

deep neural networks. Briefings in bioinformatics 22, 2096–2105.
[142] Peng, Z., Huang, W., Luo, M., Zheng, Q., Rong, Y., Xu, T., Huang, J., 2020. Graph representation learning via graphical mutual information maximization, in: WWW, pp. 259–270. URL: https://github.com/zpeng27/GMI.
[143] Perozzi, B., Al-Rfou, R., Skiena, S., 2014. Deepwalk: Online learning of social representations, in: SIGKDD, pp. 701–710. URL: https://github.com/phanein/deepwalk.
[144] Qiu, J., Chen, Q., Dong, Y., Zhang, J., Yang, H., Ding, M., Wang, K., Tang, J., 2020. Gcc: Graph contrastive coding for graph neural network pre-training, in: SIGKDD, pp. 1150–1160. URL: https://github.com/THUDM/GCC.
[145] Qu, M., Tang, J., Shang, J., Ren, X., Zhang, M., Han, J., 2017. An attention-based collaboration framework for multi-view network representation learning, in: CIKM, pp. 1767–1776. URL: https://github.com/mnqu/MVE.
[146] Rajkomar, A., Oren, E., Chen, K., Dai, A.M., Hajaj, N., Hardt, M., Liu, P.J., Liu, X., Marcus, J., Sun, M., et al., 2018. Scalable and accurate deep learning with electronic health records. NPJ Digital Medicine 1, 1–10.
[147] Roweis, S.T., Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326. URL: https://github.com/palash1992/GEM.
[148] Sankar, A., Wu, Y., Gou, L., Zhang, W., Yang, H., 2020. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks, in: WSDM, p. 519–527. URL: https://github.com/aravindsankar28/DySAT.
[149] Sanyal, S., Anishchenko, I., Dagar, A., Baker, D., Talukdar, P., 2020. Proteingcn: Protein model quality assessment using graph convolutional networks. BioRxiv .
[150] Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M., 2018. Modeling relational data with graph convolutional networks, in: ESWC, Springer. pp. 593–607. URL: https://github.com/MichSchli/RelationPrediction.
[151] Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., Tang, J., 2020. Graphaf: a flow-based autoregressive model for molecular graph generation, in: ICLR.
[152] Shi, Y., Gui, H., Zhu, Q., Kaplan, L., Han, J., 2018a. Aspem: Embedding learning by aspects in heterogeneous information networks, in: SDM, pp. 144–152. URL: https://github.com/ysyushi/aspem.
[153] Shi, Y., Zhu, Q., Guo, F., Zhang, C., Han, J., 2018b. Easing embedding learning by comprehensive transcription of heterogeneous information networks, in: SIGKDD, pp. 2190–2199. URL: https://github.com/GentleZhu/HEER.
[154] Singer, U., Guy, I., Radinsky, K., 2019. Node embedding over temporal graphs, in: Kraus, S. (Ed.), IJCAI, pp. 4605–4612. URL: https://github.com/urielsinger/tNodeEmbed.
[155] Song, X., Li, J., Lei, Q., Zhao, W., Chen, Y., Mian, A., 2022. Bi-clkt: Bi-graph contrastive learning based knowledge tracing. KBS 241, 108274.
[156] Song, X., Li, J., Tang, Y., Zhao, T., Chen, Y., Guan, Z., 2021. Jkt: A joint graph convolutional network based deep knowledge tracing. INFORM SCIENCES 580, 510–523.
[157] Steenwinckel, B., Vandewiele, G., Rausch, I., Heyvaert, P., Taelman, R., Colpaert, P., Simoens, P., Dimou, A., De Turck, F., Ongenae, F., 2020. Facilitating the analysis of covid-19 literature through a knowledge graph, in: SEMWEB, pp. 344–357.
[158] Stewart, G.W., 1990. Matrix perturbation theory .
[159] Strokach, A., Becerra, D., Corbi-Verge, C., Perez-Riba, A., Kim, P.M., 2020. Fast and flexible protein design using deep graph neural networks. CELL SYST 11, 402–411.
[160] Studer, L., Wallau, J., Dawson, H., Zlobec, I., Fischer, A., 2021. Classification of intestinal gland cell-graphs using graph neural networks, in: ICPR, IEEE. pp. 3636–3643.
[161] Su, C., Tong, J., Zhu, Y., Cui, P., Wang, F., 2020. Network embedding in biomedical data science. Briefings in bioinformatics 21, 182–197.
[162] Sun, F., Hoffmann, J., Verma, V., Tang, J., 2020. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization, in: ICLR.
[163] Sun, M., Xing, J., Wang, H., Chen, B., Zhou, J., 2021. Mocl: data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph, in: SIGKDD, pp. 3585–3594.
[164] Sun, Z., Deng, Z., Nie, J., Tang, J., 2019. Rotate: Knowledge graph embedding by relational rotation in complex space, in: ICLR. URL: https://github.com/thunlp/OpenKE.
[165] Taheri, A., Berger-Wolf, T., 2019. Predictive temporal embedding of dynamic graphs, in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 57–64.
[166] Tang, J., Qu, M., Mei, Q., 2015a. Pte: Predictive text embedding through large-scale heterogeneous text networks, in: SIGKDD, pp. 1165–1174. URL: https://github.com/mnqu/PTE.
[167] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q., 2015b. Line: Large-scale information network embedding, in: WWW.
[168] Tang, X., Luo, J., Shen, C., Lai, Z., 2021. Multi-view multichannel attention graph convolutional network for mirna–disease association prediction. Briefings in bioinformatics .
[169] Thafar, M.A., Olayan, R.S., Ashoor, H., Albaradei, S., Bajic, V.B., Gao, X., Gojobori, T., Essack, M., 2020. Dtigems+: drug–target interaction prediction using graph embedding, graph mining, and similarity-based techniques. Jnl of Cheminformatics 12, 1–17.
[170] Toivonen, H., Srinivasan, A., King, R.D., Kramer, S., Helma, C., 2003. Statistical evaluation of the predictive toxicology challenge 2000–2001. Bioinformatics 19, 1183–1193.
[171] Torng, W., Altman, R.B., 2019. Graph convolutional neural networks for predicting drug-target interactions. J CHEM INF MODEL 59, 4131–4149.
[172] Trivedi, P., Lubana, E.S., Yan, Y., Yang, Y., Koutra, D., 2022. Augmentations in graph contrastive learning: Current methodological flaws & towards better practices, in: WWW, pp. 1538–1549. URL: https://github.com/GemsLab/GCLAugmentations-BestPractices.
[173] Trivedi, R., Dai, H., Wang, Y., Song, L., 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs, in: ICML, pp. 3462–3471. URL: https://github.com/rstriv/Know-Evolve.
[174] Trivedi, R., Farajtabar, M., Biswal, P., Zha, H., 2019. Dyrep: Learning representations over dynamic graphs, in: ICLR.
[175] Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G., 2016. Complex embeddings for simple link prediction, in: ICML, pp. 2071–2080. URL: https://github.com/thunlp/OpenKE.
[176] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv:1710.10903 URL: https://github.com/PetarV-/GAT.
[177] Velickovic, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D., 2019. Deep graph infomax. ICLR 2, 4. URL: https://github.com/PetarV-/DGI.
[178] Wan, F., Hong, L., Xiao, A., Jiang, T., Zeng, J., 2019. Neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions. Bioinformatics 35.
[179] Wang, D., Cui, P., Zhu, W., 2016. Structural deep network embedding, in: SIGKDD, pp. 1225–1234. URL: https://github.com/suanrong/SDNE.
[180] Wang, K., Liu, Y., Sheng, Q.Z., 2022a. Swift and sure: Hardness-aware contrastive learning for low-dimensional knowledge graph embeddings, in: WWW, pp. 838–849.
[181] Wang, Q., Mao, Z., Wang, B., Guo, L., 2017. Knowledge graph embedding: A survey of approaches and applications. TKDE 29, 2724–2743.
[182] Wang, S., Cho, H., Zhai, C., Berger, B., Peng, J., 2015. Exploiting ontology graph for predicting sparsely annotated gene function. Bioinformatics 31, i357–i364.
[183] Wang, S., Huang, E., Cairns, J., Peng, J., Wang, L., Sinha, S., 2019a. Identification of pathways associated with chemosensitivity through network embedding. PLOS COMPUT BIOL 15, e1006864.
[184] Wang, S.H., Govindaraj, V.V., Górriz, J.M., Zhang, X., Zhang, Y.D., 2021a. Covid-19 classification by fgcnet with deep feature fusion

from graph convolutional network and convolutional neural network. Inf Fusion 67, 208–229.

[185] Wang, X., Bo, D., Shi, C., Fan, S., Ye, Y., Philip, S.Y., 2022b. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. IEEE Transactions on Big Data .

[186] Wang, X., Gong, Y., Yi, J., Zhang, W., 2019b. Predicting gene-disease associations from the heterogeneous network using graph embedding, in: BIBM, pp. 504–511.

[187] Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S., 2019c. Heterogeneous graph attention network, in: WWW, pp. 2022–2032. URL: https://github.com/Jhy1993/HAN.

[188] Wang, X., Liu, N., Han, H., Shi, C., 2021b. Self-supervised heterogeneous graph neural network with co-contrastive learning, in: SIGKDD. URL: https://github.com/liun-online/HeCo.

[189] Wang, X., Zhang, Y., Shi, C., 2019d. Hyperbolic heterogeneous information network embedding, in: AAAI, pp. 5337–5344. URL: https://github.com/ydzhang-stormstout/HHNE.

[190] Wang, Z., Zhang, J., Feng, J., Chen, Z., 2014. Knowledge graph embedding by translating on hyperplanes, in: AAAI. URL: https://github.com/thunlp/OpenKE.

[191] Wei, H., Hu, G., Bai, W., Xia, S., 2019. Lifelong representation learning in dynamic attributed network. Neurocomputing .

[192] Wise, C., Ioannidis, V.N., Calvo, M.R., Song, X., Price, G., Kulkarni, N., Brand, R., Bhatia, P., Karypis, G., 2020. Covid-19 knowledge graph: accelerating information retrieval and discovery for scientific literature. arXiv:2007.12731 .

[193] Wu, T., Wang, Y., Wang, Y., Zhao, E., Yuan, Y., 2021. Leveraging graph-based hierarchical medical entity embedding for healthcare applications. Sci. Rep. 11, 1–13.

[194] Xia, F., Sun, K., Yu, S., Aziz, A., Wan, L., Pan, S., Liu, H., 2021. Graph learning: A survey. IEEE Trans. Artif. Intell. 2, 109–127.

[195] Xia, J., Wu, L., Chen, J., Hu, B., Li, S.Z., 2022. Simgrace: A simple framework for graph contrastive learning without data augmentation, in: WWW, pp. 1070–1079. URL: https://github.com/junxia97/simgrace.

[196] Xinyi, Z., Chen, L., 2019. Capsule graph neural network, in: ICLR.

[197] Xu, C., Guan, Z., Zhao, W., Niu, Y., Wang, Q., Wang, Z., 2018. Deep multi-view concept learning., in: IJCAI, pp. 2898–2904.

[198] Xu, L., Wei, X., Cao, J., Yu, P.S., 2017. Embedding of embedding (eoe) joint embedding for coupled heterogeneous networks, in: WSDM, pp. 741–749.

[199] Xuan, P., Pan, S., Zhang, T., Liu, Y., Sun, H., 2019a. Graph convolutional network and convolutional neural network based method for predicting lncrna-disease associations. Cells 8, 1012.

[200] Xuan, P., Sheng, N., Zhang, T., Liu, Y., Guo, Y., 2019b. A method based on convolutional autoencoder and convolutional neural network with adjacent edge attention for predicting lncrna–disease associations. Int. J. Mol. Sci. 20, 4260.

[201] Xuan, P., Sun, H., Wang, X., Zhang, T., Pan, S., 2019c. Inferring the disease-associated mirnas based on network representation learning and convolutional neural networks. Int. J. Mol. Sci. 20, 3648.

[202] Xue, G., Zhong, M., Li, J., Chen, J., Zhai, C., Kong, R., 2022. Dynamic network embedding survey. Neurocomputing 472, 212–223.

[203] Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W., Kanehisa, M., 2008. Prediction of drug–target interaction networks from integration of chemical and genomic spaces. Bioinformatics 24, i232–i240.

[204] Yang, B., Yih, W., He, X., Gao, J., Deng, L., 2015a. Embedding entities and relations for learning and inference in knowledge bases, in: ICLR. URL: https://github.com/thunlp/OpenKE.

[205] Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E., 2015b. Network representation learning with rich text information, in: IJCAI. URL: https://github.com/thunlp/tadw.

[206] Yang, H., Pan, S., Zhang, P., Chen, L., Lian, D., Zhang, C., 2018. Binarized attributed network embedding, in: ICDM, pp. 1476–1481.

[207] Yang, J., Li, Z., Fan, X., Cheng, Y., 2014. Drug–disease association and drug-repositioning predictions in complex diseases using causal inference–probabilistic matrix factorization. Jnl of chemical information and modeling 54, 2562–2569.

[208] Yang, L., Xiao, Z., Jiang, W., Wei, Y., Hu, Y., Wang, H., 2020. Dynamic heterogeneous graph embedding using hierarchical attentions, in: ECIR, pp. 425–432.

[209] Yao, H., Guan, J., Liu, T., 2020. Denoising protein–protein interaction network via variational graph auto-encoder for protein complex detection. JBCB 18.

[210] Yi, H.C., You, Z.H., Huang, D.S., Kwoh, C.K., 2021. Graph representation learning in bioinformatics: trends, methods and applications. Briefings in bioinformatics .

[211] Yin, Y., Wang, Q., Huang, S., Xiong, H., Zhang, X., 2022. Autogcl: Automated graph contrastive learning via learnable view generators, in: AAAI, pp. 8892–8900. URL: https://github.com/Somedaywilldo/AutoGCL.

[212] You, R., Yao, S., Mamitsuka, H., Zhu, S., 2021a. Deepgraphgo: graph neural network for large-scale, multispecies protein function prediction. Bioinformatics 37, i262–i271.

[213] You, Y., Chen, T., Shen, Y., Wang, Z., 2021b. Graph contrastive learning automated, in: ICML, pp. 12121–12132. URL: https://github.com/Shen-Lab/GraphCL_Automated.

[214] You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y., 2020. Graph contrastive learning with augmentations. NeurIPS 33, 5812–5823. URL: https://github.com/Shen-Lab/GraphCL.

[215] You, Z.H., Lei, Y.K., Gui, J., Huang, D.S., Zhou, X., 2010. Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data. Bioinformatics 26.

[216] Yu, G., Wang, K., Fu, G., Guo, M., Wang, J., 2018a. Nmfgo: gene function prediction via nonnegative matrix factorization with gene ontology. TCBB 17, 238–249.

[217] Yu, G., Zhu, H., Domeniconi, C., Liu, J., 2015. Predicting protein function via downward random walks on a gene ontology. BMC Bioinform. 16, 1–13.

[218] Yu, W., Aggarwal, C.C., Wang, W., 2017. Temporally factorized network modeling for evolutionary network analysis, in: WSDM.

[219] Yu, W., Cheng, W., Aggarwal, C.C., Zhang, K., Chen, H., Wang, W., 2018b. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks, in: SIGKDD, pp. 2672–2681. URL: https://github.com/chengw07/NetWalk.

[220] Yu, W., Zheng, C., Cheng, W., Aggarwal, C.C., Song, D., Zong, B., Chen, H., Wang, W., 2018c. Learning deep network representations with adversarially regularized autoencoders, in: SIGKDD, pp. 2663–2671. URL: https://github.com/jakezhaojb/ARAE.

[221] Yu, Z., Huang, F., Zhao, X., Xiao, W., Zhang, W., 2021. Predicting drug–disease associations through layer attention graph convolutional network. Briefings in bioinformatics 22, bbaa243.

[222] Yue, X., Wang, Z., Huang, J., Parthasarathy, S., Moosavinasab, S., Huang, Y., Lin, S.M., Zhang, W., Zhang, P., Sun, H., 2020. Graph embedding on biomedical networks: methods, applications and evaluations. Bioinformatics 36, 1241–1251.

[223] Yun, S., Jeong, M., Kim, R., Kang, J., Kim, H.J., 2019. Graph transformer networks, in: NeurIPS, pp. 11960–11970. URL: https://github.com/seongjunyun/Graph_Transformer_Networks.

[224] Zang, C., Wang, F., 2020. Moflow: an invertible flow model for generating molecular graphs, in: SIGKDD, pp. 617–626.

[225] Zeng, Y., Zhou, X., Rao, J., Lu, Y., Yang, Y., 2020. Accurately clustering single-cell rna-seq data by capturing structural relations between cells through gnn, in: BIBM, pp. 519–522.

[226] Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V., 2019a. Heterogeneous graph neural network, in: SIGKDD, pp. 793–803. URL: https://github.com/chuxuzhang/KDD2019_HetGNN.

[227] Zhang, D., Yin, J., Zhu, X., Zhang, C., 2018a. Complex semantic path augmented heterogeneous network embedding, in: PAKDD, pp. 196–208. URL: https://github.com/daokunzhang/MetaGraph2Vec.

[228] Zhang, J., Hu, X., Jiang, Z., Song, B., Quan, W., Chen, Z., 2019b. Predicting disease-related rna associations based on graph convolutional attention network, in: BIBM, pp. 177–182.

[229] Zhang, J., Jiang, Z., Hu, X., Song, B., 2020a. A novel graph attention adversarial network for predicting disease-related associations. Methods 179, 81–88.

[230] Zhang, J., Shi, X., Xie, J., Ma, H., King, I., Yeung, D., 2018b. Gaan: Gated attention networks for learning on large and spatiotemporal graphs, in: UAI, pp. 339–349.
[231] Zhang, L., Liu, B., Li, Z., Zhu, X., Liang, Z., An, J., 2020b. Predicting mirna-disease associations by multiple meta-paths fusion graph embedding model. BMC Bioinform. 21, 1–19.
[232] Zhang, M., Dong, B., Li, Q., 2022. Ms-gwnn: multi-scale graph wavelet neural network for breast cancer diagnosis, in: ISBI, IEEE.
[233] Zhang, S., Tay, Y., Yao, L., Liu, Q., 2019c. Quaternion knowledge graph embeddings, in: NeurIPS, pp. 2731–2741.
[234] Zhang, W., Fang, Y., Liu, Z., Wu, M., Zhang, X., 2020c. Learning relationship-preserving heterogeneous graph representations via metagraph embedding. TKDE URL: https://github.com/zwtywwq/mg2vec.
[235] Zhang, W., Yue, X., Lin, W., Wu, W., Liu, R., Huang, F., Liu, F., 2018c. Predicting drug-disease associations by using similarity constrained matrix factorization. BMC Bioinform. 19, 1–12.
[236] Zhang, Z., Cui, P., Pei, J., Wang, X., Zhu, W., 2018d. Timers: Error-bounded svd restart on dynamic networks, in: AAAI. URL: zhang2018timers.
[237] Zhao, C., Jiang, J., Guan, Y., Guo, X., He, B., 2018. Emr-based medical knowledge representation and inference via markov random fields and distributed representation learning. AIM 87, 49–59.
[238] Zhao, C., Qiu, Y., Zhou, S., Liu, S., Zhang, W., Niu, Y., 2020a. Graph embedding ensemble methods based on the heterogeneous network for lncrna-mirna interaction prediction. BMC genomics 21.
[239] Zhao, J., Wang, X., Shi, C., Liu, Z., Ye, Y., 2020b. Network schema preserving heterogeneous information network embedding, IJCAI. URL: https://github.com/Andy-Border/NSHE.
[240] Zhao, K., Bai, T., Wu, B., Wang, B., Zhang, Y., Yang, Y., Nie, J.Y., 2020c. Deep adversarial completion for sparse heterogeneous information network embedding, in: WWW, pp. 508–518.
[241] Zhao, T., Hu, Y., Valsdottir, L.R., Zang, T., Peng, J., 2021. Identifying drug–target interactions based on graph convolutional network and deep neural network. Briefings in bioinformatics 22.
[242] Zhong, X., Rajapakse, J.C., 2020. Graph embeddings on gene ontology annotations for protein-protein interaction prediction. BMC Bioinform. 21-S, 560.
[243] Zhou, H., He, L., Zhang, Y., Shen, L., Chen, B., 2022. Interpretable graph convolutional network of multi-modality brain imaging for alzheimer's disease diagnosis, in: ISBI, IEEE. pp. 1–5.
[244] Zhou, Y., Liu, W., Pei, Y., Wang, L., Zha, D., Fu, T., 2019. Dynamic network embedding by semantic evolution, in: IJCNN.
[245] Zhu, D., Cui, P., Wang, D., Zhu, W., 2018a. Deep variational network embedding in wasserstein space, in: SIGKDD, pp. 2827–2836. URL: https://github.com/Shen-Lab/Drug-Combo-Generator.
[246] Zhu, D., Cui, P., Zhang, Z., Pei, J., Zhu, W., 2018b. High-order proximity preserved embedding for dynamic networks. TKDE 30, 2134–2144. URL: https://github.com/ZW-ZHANG/DHPE.
[247] Zhu, L., Guo, D., Yin, J., Ver Steeg, G., Galstyan, A., 2016. Scalable temporal latent space inference for link prediction in dynamic social networks. TKDE 28, 2765–2777.
[248] Zhu, L., Hong, Z., Zheng, H., 2019a. Predicting gene-disease associations via graph embedding and gnns, in: BIBM, pp. 382–389.
[249] Zhu, L., You, Z.H., Huang, D.S., 2013. Increasing the reliability of protein–protein interaction networks via non-convex semantic embedding. Neurocomputing 121, 99–107.
[250] Zhu, S., Zhou, C., Pan, S., Zhu, X., Wang, B., 2019b. Relation structure-aware heterogeneous graph neural network, in: ICDM, pp. 1534–1539. URL: https://github.com/CheriseZhu/RSHN.
[251] Zhu, Y., Elemento, O., Pathak, J., Wang, F., 2019c. Drug knowledge bases and their applications in biomedical informatics research. Briefings in bioinformatics 20, 1308–1321.
[252] Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L., 2020. Deep graph contrastive representation learning. arXiv:2006.04131 URL: https://github.com/CRIPAC-DIG/GRACE.
[253] Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L., 2021. Graph contrastive learning with adaptive augmentation, in: WWW, pp. 2069–2080. URL: https://github.com/CRIPAC-DIG/GCA.
[254] Zitnik, M., Agrawal, M., Leskovec, J., 2018. Modeling polypharmacy side effects with gcns. Bioinformatics 34, i457–i466.
[255] Zitnik, M., Leskovec, J., 2017. Predicting multicellular function through multi-layer tissue networks. Bioinformatics 33, i190–i198.
[256] Zitnik, M., Zupan, B., 2016. Collective pairwise classification for multi-way analysis of disease and drug data, in: PSB, pp. 81–92.
[257] Zuo, Y., Liu, G., Lin, H., Guo, J., Hu, X., Wu, J., 2018. Embedding temporal network via neighborhood formation, in: SIGKDD, pp. 2857–2866. URL: https://github.com/rmitbggroup/ICDE2020TemporalNetworkEmbedding.