Chapter 2
The State-of-the-art of Heterogeneous Graph Representation

Abstract In this chapter, we give a comprehensive review of the recent development on heterogeneous graph representation (HGR) methods and techniques. In the method aspect, according to the information used in HGR, existing works are divided into four categories, i.e., structure-preserved HGR, attribute-assisted HGR, dynamic HGR and application-oriented HGR. In the technique aspect, we summary five commonly used techniques in HGR and categorize them into shallow model and deep model. In addition, we also provide some public sources, e.g., benchmark datasets, source code and available tools.

2.1 Method Taxonomy

Various types of nodes and links in HG bring complex graph structures and rich attributes, i.e., the heterogeneity of HG. To make the node representation capture the heterogeneity, we need to consider the information of different aspects, including graph structures, attributes, specific domain knowledge and so on. In this chapter, we categorize the existing methods into four categories based on the information they used in HGR. An overview of existing HGR methods explored in this book is shown in Fig. 2.1.

2.1.1 Structure-preserved Representation

One basic requirement of graph representation is to preserve the graph structures properly [7]. For example, in the homogeneous graph representation, existing works consider a lot of graph structures, e.g., first-order structure [37], second-order structure [47, 50], high-order structure [1, 71] and community structure [53]. Due to the heterogeneity of HG, the graph structures become more complex, and even have semantic information, e.g., the co-author relationship. Therefore, an important di-

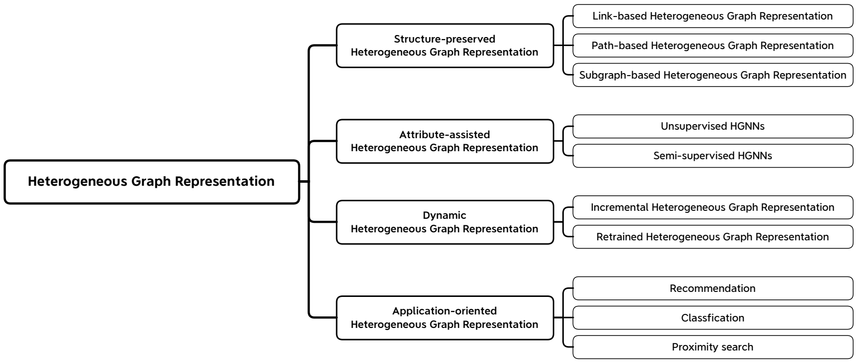
Heterogeneous Graph Representation
Structure-preserved Heterogeneous Graph Representation
Link-based Heterogeneous Graph Representation
Path-based Heterogeneous Graph Representation
Subgraph-based Heterogeneous Graph Representation
Attribute-assisted Heterogeneous Graph Representation
Unsupervised HGNNs
Semi-supervised HGNNs
Dynamic Heterogeneous Graph Representation
Incremental Heterogeneous Graph Representation
Retrained Heterogeneous Graph Representation
Application-oriented Heterogeneous Graph Representation
Recommendation
Classfication
Proximity search

Fig. 2.1 An overview of heterogeneous graph representation methods.

rection of HGR is to learn both the structural and semantic information simultaneously. In this chapter, we review the typical structured-preserved HGR methods. Each of them considers different structures in HG, including link/edge, meta-path, and subgraph.

A basic requirement of HGR is to preserve the multiple relationships, i.e., links, in node representations. Different from homogeneous graph, links in HG have different types and semantics. To distinguish various types of links, one classical idea is to project them into different metric spaces, rather than a unified metric space. A representative work of this idea is PME [4], which treats each link type as a relation and uses a relation-specific matrix to transform the nodes into different metric spaces. In this way, nodes connected by different types of links can be close to each other in different metric spaces, thus capturing the heterogeneity of the graph. Different from PME, EOE [59] and HeGAN [21] use the relation-specific matrix to calculate the similarity between two nodes. AspEM [43] and HEER [44] aim to maximize the probability of existing links. Generally, the key point of designing link-based methods is to find a proper heterogeneous similarity function to preserve the proximity between nodes.

Link-based methods can only capture the local structures of HG, i.e., the first-order information. In fact, the higher-order relation, describing more complex semantic information (e.g., the co-author relationship), is also critical for HGR. Meta-path is a commonly used tool in modeling the high-order relationship of HG. A representative work is metapath2vec [8], which uses metapath-guided random walk to generate heterogeneous node sequences with rich semantics; and then it designs a heterogeneous skip-gram technique to preserve the proximity between nodes and their context nodes. Based on metapath2vec, a series of variants have been proposed. For example, Spacey [17] designs a heterogeneous spacey random walk to unify different meta-paths into a second-order hyper-matrix. JUST [26] proposes a random walk method with Jump and Stay strategies, which can flexibly choose to change or maintain the type of the next node in the random walk without meta-path. BHIN2vec [29] designs an extended skip-gram technique to balance the various types of relations. HHNE [57] conducts the metapath-guided random walk in hyperbolic spaces [18]. Besides, HEAD [52] separates the node representations into intrinsic representations and meta-path specific representations, so that the highly coupled representations can be well disentangled and become more robust.

Subgraph represents a more complex structure of HG. Incorporating subgraphs into graph representation can significantly improve the ability of capturing complex structural relationship. Zhang *et al.* propose metagraph2vec [67], which uses a metagraph-guided random walk to generate heterogeneous node sequence. Then a heterogeneous skip-gram technique [8] is employed to learn the node representations. Based on this strategy, metagraph2vec can capture the high-order similarity and rich semantic information between nodes. DHNE [49] is a hyperedge-based HGR method. Specifically, it designs a novel deep model to produce a non-linear tuple-wise similarity function while capturing the local and global structures of a given HG.

Compared with link and meta-path, subgraph usually contains more higher-order structural and semantic information. However, one obstacle of subgraph-based HGR methods is the high complexity of subgraph. Therefore, how to balance the effectiveness and efficiency is required for a practical subgraph-based HGR methods, which is worthy of further exploration.

2.1.2 Attribute-assisted Representation

In addition to the graph structures, another important component of HGR is the rich attributes. Attribute-assisted HGR methods, i.e., heterogeneous graph neural networks (HGNNs), aim to encode the complex structures and rich attributes together to learn node representations. Different from graph neural networks (GNNs) that can directly fuse the attributes of neighbors to update node representations, HGNNs need to overcome the heterogeneity of node/edge attributes and design effective fusion mechanisms to utilize the neighborhood information, which is more challenging. In this chapter, we divide HGNNs into unsupervised and semi-supervised settings, and discuss their own pros and cons.

The goal of unsupervised HGNNs is to learn node representations that benefit downstream tasks in an unsupervised manner. To this end, they usually utilize the interactions between different types of nodes/edges to capture the potential proximity, so that the learned representation can have good generalization.

HetGNN [65] is the representative work of unsupervised HGNNs. It consists of three parts: content aggregation, neighbor aggregation and type aggregation. Content aggregation is designed to fuse the multiple attributes in a node, e.g., a film can have image and text attributes simultaneously. Neighbor aggregation aims to aggregate the nodes with same type. And type aggregation uses an attention mechanism to mix the representations of different types and produces the final node representations. Through these three components, HetGNN can preserve the heterogeneity of both graph structures and node attributes. Some other unsupervised methods can be regarded as special cases of HetGNN because they either capture the heterogeneity of node attributes or the heterogeneity of graph structures. HNE [3] is proposed to learn representations for the cross-model data in HG, but it ignores the various types of links. SHNE [66] focuses on capturing the semantic information of nodes by designing a deep semantic encoder with gated recurrent units (GRU) [6]. Although it uses heterogeneous skip-gram to preserve the heterogeneity of graph, SHNE is designed only for text data.

Besides, GATNE [2] aims to learn node representations in multiplex graph, i.e., a heterogeneous graph with different types of edges. Therefore, it pays more attention to distinguish different link relationships. HeCo [55] uses self-supervised learning, i.e., contrastive learning, to generate supervised signals. It designs a novel co-contrastive mechanism to capture the meta-path information and network schema information simultaneously.

It can be seen that the purpose of unsupervised HGNNs is to save as much information as possible. For example, HetGNN uses three types of aggregation functions to learn the information of content, neighbor and node type separately. HeCo captures the information of meta-path and network schema. The reason is that the representations learned by unsupervised HGNNs need to be used for downstream tasks, so that it should cover the information of different aspects.

Different from unsupervised HGNNs, semi-supervised HGNNs aim to learn task-specific node representations. Therefore, they prefer to use attention mechanism to capture the most relevant structural and attribute information. Wang *et al.* [54] propose heterogeneous graph attention network (HAN), which uses a hierarchical attention mechanism to capture both node and semantic importance. Then a series of attention-based HGNNs were proposed [12, 19, 25, 13]. MAGNN [12] designs intra-metapath aggregation and inter-metapath aggregation. HetSANN [19] and HGT [25] use self-attention mechanism, which treats one type of node as query to calculate the importance of other types of nodes around it. [13] uses meta-paths as virtual edges to enhance the performance of graph attention operator.

Compared with structure-preserved HGR methods, HGNNs have an obvious advantage that they have the ability of inductive learning, i.e., learning representations for the out-of-sample nodes [20]. Besides, HGNNs need less memory space because they only need to store model parameters. These two reasons are important for the real-world applications. However, they still suffer from the huge time costing in inference and retraining.

2.1.3 Dynamic Representation

The real-world graphs are constantly changing over time. For example, in the social platform, people follow and unfollow others daily. Therefore, capturing the temporal information of HG is an important research direction. In this chapter, we introduce the typical dynamic HGR methods, which can be divided into two categories: incremental update and retrained update methods. The former learns the representation of new node in the next timestamp by utilize existing node representations, while the latter retrains the models in each timestamp.

DyHNE [56] is an incremental update method based on the theory of matrix perturbation, which aims to learn node representations, and considers both the heterogeneity and evolution of HG at the same time. DyHNE first preserves the meta-path based first- and second-order proximity. Then it uses the perturbation of meta-path augmented adjacency matrices to capture the changes of graph. Besides, some methods try to use GNNs to learn node or edge representations in each timestamp and then design some advanced neural network, e.g., RNN or attention mechanism, to capture the temporal information of HG. DyHATR [61] aims to capture the temporal information through the changes of nodes representations in different timestamps. To this end, it first designs a hierarchical attention mechanism (HAT), which contains a node- and edge-level attention, to learn node representations by fusing the attributes of neighbors.

It can be seen that the incremental update methods are efficient, but they can only capture the short-term temporal information, i.e., the last timestamp [61]. Besides, incremental update methods focus on utilizing non-linear model, and lack expressive power. On the contrary, the retrained update methods employ neural networks to capture the long-term temporal information. However, they suffer from the high computational cost. Therefore, how to combine the advantages of these two kinds of models is an important problem.

2.1.4 Application-oriented Representation

HGR can be integrated with some specific applications. In this situation, one usually needs to consider two factors: one is how to construct an HG for a specific application, another is what information, i.e., domain knowledge, should be incorporated into HGR. Here, we discuss three common types of applications: recommendation, classification and proximity search.

Recommendation can be naturally modeled as a link prediction task on HG, where there are at least two types of nodes to represent users and items separately, and links represent the interaction between them. Therefore, HGR is widely used in the recommendation scenario [41]. Besides, other types of information, such as the social relationships, can be easily applied to HG [42], so applying HGR to recommendation application is an important research field.

HERec [39] aims to learn the representations of users and items under different meta-paths and fuse them for recommendation. It first finds the co-occurrence of users and items based on the metapath-guided random walks on user-item HG. Then it uses node2vec [14] to learn preliminary representation from node sequences of users and items. Because the representations under different meta-paths contain different semantic information, for better recommendation performance, HERec designs a fusion function to unify the multiple representations. Apart from random walk, some methods try to use matrix factorization to learn user and item representations. HeteRec [62] considers the implicit user feedback in HG. HeteroMF [27] designs a heterogeneous matrix factorization technique to consider the context dependence of different types of nodes. FMG [72] incorporates meta-graph into HGR, which can capture some special patterns between users and items.

Previous methods mainly use non-linear model to learn the representations of users and items, which cannot fully capture users' preferences. Therefore, some neural network-based methods are proposed. One of the most important techniques is attention mechanism. MCRec [22] designs a neural co-attention mechanism to capture the relationship between user, item and meta-path. NeuACF [16] and HueRec [58] first calculate multiple metapath-based user-item proximity matrices. Then an attention mechanism is designed to learn the importance of different proximity matrices, which learns the importance of users' preferences.

Another type of methods is to apply HGNNs to recommendation. PGCN [60] converts the user-item interaction sequences into item-item graph, user-item graph and user-sequence graph. Then it designs an HGNN to aggregate the information of user and item in the three graphs, thus capturing the collaborative filtering signals. SHCF [30] uses HGNNs to capture both the high-order heterogeneous collaborative signals and sequential information simultaneously. GNewsRec [23] and GNUD [24] are designed for news recommendation. They consider both the content information of news and the collaborative information between users and news.

Classification is a fundamental task in machine learning. Here we mainly introduce two types of classification tasks that require models to capture the heterogeneity of HG: author identification [5, 64, 36] and user identification [70, 10, 69].

Author identification aims to find the potential authors for an anonymous paper in the academic network. Camel [64] is designed to consider both the content information, e.g., the text of papers, and context information, e.g., the co-occurrence of paper, author and conference. PAHNE [5] uses meta-paths to augment the pair-wise relations between paper and author. TaPEm [36] further maximizes the proximity between the paper-author pair and the context path around them.

User identification requires the model to make use of the heterogeneity of HG to learn discriminating user representations with weak supervision information. Player2vec [70], AHIN2vec [10] and Vendor2vec [69] are the principal methods. They can be summarized as a general framework: first, some advanced neural networks are used to learn preliminary node representations from the input features. Then the representations will be propagated on the constructed HG to capture the heterogeneity of HG. Finally, a semi-supervised loss function is used to make the node representations contain application-specific information. Under the guidance of partially labeled nodes, the node representations can distinguish special users from the ordinary users in the graph, which can be used for user identification.

Proximity search aims to find the nodes that are closest to the target node by using structural and semantic information of HG. Some earlier studies have deal with this problem in homogeneous graphs, for example, web search [28]. Recently, some methods try to utilize HG in proximity search [45, 40]. However, these methods only use some statistical information, e.g., the number of connected meta-paths, to measure the similarity of two nodes in HG, which lack flexibility. With the development of deep learning, some graph representation methods are proposed. IPE [31] considers the interactions among different meta-path instances and proposes an interactive-paths structure to improve the performance of HGR. SPE [32] proposes a subgraph-augmented HGR method, which uses a stacked autoencoder to learn the subgraph representation so as to enhance the effect of semantic proximity search. D2AGE [33] explores the directed acyclic graph (DAG) structure for better measuring the similarity between two nodes and designs a DAG-LSTM to learn node representations.

In summary, incorporating HGR into specific applications usually needs to consider the domain knowledge. For example, in recommendation, meta-path "user-item-user" can be used to capture the user-based collaborative filtering, while "item-user-item" represents the item-based collaborative filtering; in proximity search, the methods use meta-paths to capture the semantic relationships between nodes, thus enhancing the performance. Therefore, utilizing HG to capture the application-specific domain knowledge is essential for application-oriented HGR.

2.2 Technique Summary

In this chapter, from the technical perspective, we summarize the widely used techniques (or models) in HGR, which can be generally divided into two categories: shallow model and deep model.

2.2.1 Shallow Model

Early HGR methods focus on employing shallow model. They first initialize the node representations randomly, and then learn the node representations through optimizing some well-designed objective functions. We divide the shallow model into two categories: random walk-based and decomposition-based.

Random walk-based. In homogeneous graph, random walk, which generates some node sequences in a graph, is usually used to capture the local structure of a graph [14]. While in heterogeneous graph, the node sequence should contain not only the structural information, but also the semantic information. Therefore, a series of semantic-aware random walk techniques are proposed [68, 8, 17, 26, 29, 57, 39]. For example, metapath2vec [8] uses meta-path-guided random walk to capture the semantic information of two nodes, e.g., the co-author relationship in academic graph. Spacey [17] and metagraph2vec [67] design metagraph-guided random walks, which preserve a more complex similarity between two nodes.

Decomposition-based. Decomposition-based techniques aim to decompose HG into several sub-graphs and preserve the proximity of nodes in each sub-graph [4, 59, 43, 44, 35, 46, 15]. PME [4] decomposes the heterogeneous graph into some bipartite graphs according to the types of links and projects each bipartite graph into a relation-specific semantic space. PTE [46] divides the documents into word-word graph, word-document graph and word-label graph. Then it uses LINE [47] to learn the shared node representations for each sub-graph. HEBE [15] samples a series of subgraphs from an HG and preserves the proximity between the center node and its subgraph.

2.2.2 Deep Model

Deep model aims to use advanced neural networks to learn representation from the node attributes or the interactions among nodes, which can be roughly divided into three categories: message passing-based, encoder-decoder-based and adversarial-based.

Message passing-based. The idea of message passing is to send the node representation to its neighbors, which is always used in GNNs. The key component of message passing-based techniques is to design a suitable aggregation function, which can capture the semantic information of HG [54, 12, 19, 65, 2, 74, 63, 77, 38]. For example, HAN [54] designs a hierarchical attention mechanism to learn the importance of different nodes and meta-paths, which captures both structural information and semantic information of HG. HetGNN [65] uses bi-LSTM to aggregate the representation of neighbors so as to learn the deep interactions among heterogeneous nodes. GTN [63] designs an aggregation function, which can find the suitable meta-paths automatically during the process of message passing.

Encoder-decoder-based. Encoder-decoder-based techniques aim to employ some neural networks as encoder to learn representation from node attributes and design a decoder to preserve some properties of the graphs [49, 3, 66, 5, 64, 36]. For example, HNE [3] focuses on multi-modal heterogeneous graph. It uses CNN and autoencoder to learn representation from images and texts, respectively. Then it uses the representation to predict whether there is a link between the images and texts. Camel [64] uses GRU as encoder to learn paper representation from the abstracts. A skip-gram objective function is used to preserve the local structures of the graphs. DHNE [49] uses autoencoder to learn representation for the nodes in a hyperedge. Then it designs a binary classification loss to preserve the indecomposability of the hyper-graph.

Adversarial-based. Adversarial-based techniques utilize the game between generator and discriminator to learn robust node representation. In homogeneous graph, the adversarial-based techniques only consider the structural information, for example, GraphGAN [51] uses Breadth First Search when generating virtual nodes. In a heterogeneous graph, the discriminator and generator are designed to be relation-aware, which captures the rich semantics on HGs. HeGAN [22] is the first to use GAN in HGR. It incorporates the multiple relations into the generator and discriminator, so that the heterogeneity of a given graph can be considered. MV-ACM [76] uses GAN to generate the complementary views by computing the similarity of nodes in different views.

2.3 Open Sources

In this chapter, we summarize the commonly used datasets of HGR. In addition, we will introduce some useful resources and open-source tools about HGR.

2.3.1 Benchmark Datasets

High-quality datasets are essential for academic research. Here, we introduce some popular real-world HG datasets, which can be divided into three categories: academic networks, business networks and film networks.

- **DBLP**[2] This is a network that reflects the relationship between authors and papers. There are four types of nodes: author, paper, term and venue.
- **Aminer**[3] This academic network is similar to DBLP, but with two additional node types: keyword and conference.
- **Yelp**[4] This is a social media network, including five types of nodes: user, business, compliment, city and category.
- **Amazon**[5] This is an E-commercial network, which records the interactive information between users and products, including co-viewing, co-purchasing, etc.
- **IMDB**[6] This is a film rating network, recording the preferences of users on different films. Each film contains its directors, actors and genre.
- **Douban**[7] This network is similar to IMDB, but it contains more user information, such as the group and location of the users.

2.3.2 Open-source Code

Source code is important for researchers to reproduce the corresponding method. In Table 2.1, we refer to the related papers of the datasets. Furthermore, we collect the source code of the related papers and list them in Table 2.1. Besides, we provide some commonly used website about graph representation.

- Stanford Network Analysis Project (SNAP). It is a network analysis and graph mining library, which contains different types of networks and multiple network analysis tools. The address is http://snap.stanford.edu/.

[2] http://dblp.uni-trier.de

[3] https://www.aminer.cn

[4] http://www.yelp.com/dataset challenge/

[5] http://jmcauley.ucsd.edu/data/amazon

[6] https://grouplens.org/datasets/movielens/100k/

[7] http://movie.douban.com/

Table 2.1 Source code of related papers.

Method	Source code	Programing platform
metapath2vec [8]	https://github.com/apple2373/metapath2vec	Tensorflow
metagraph2vec [67]	https://github.com/daokunzhang/MetaGraph2Vec	C++
AspEM [43]	https://github.com/ysyushi/aspem	Python
HEER [44]	https://github.com/GentleZhu/HEER	Python
HEBE [15]	https://github.com/olittle/Hebe	C++
JUST [26]	https://github.com/eXascaleInfolab/JUST	Python
HIN2vec [11]	https://github.com/csiesheep/hin2vec	Python & C++
BHIN2vec [29]	https://github.com/sh0416/BHIN2VEC	Pytorch
HHNE [57]	https://github.com/ydzhang-stormstout/HHNE	C++
HeRec [39]	https://github.com/librahu/HERec	Python
MNE [68]	https://github.com/HKUST-KnowComp/MNE	Python
PTE [46]	https://github.com/mnqu/PTE	C++
RHINE [34]	https://github.com/rootlu/RHINE	Pytorch
HAN [54]	https://github.com/Jhy1993/HAN	Tensorflow
MAGNN [12]	https://github.com/cynricfu/MAGNN	Pytorch
HetSANN [19]	https://github.com/didi/hetsann	Tensorflow
HGT [25]	https://github.com/acbull/pyHGT	Pytorch
HetGNN [65]	https://github.com/chuxuzhang/KDD2019_HetGNN	Pytorch
GATNE [2]	https://github.com/THUDM/GATNE	Pytorch
RSHN [77]	https://github.com/CheriseZhu/RSHN	Pytorch
RGCN [38]	https://github.com/tkipf/relational-gcn	Tensorflow
IntentGC [75]	https://github.com/peter14121/intentgc-models	Python
MEIRec [9]	https://github.com/googlebaba/KDD2019-MEIRec	Tensorflow
GNUD [24]	https://github.com/siyongxu/GNUD	Tensorflow
FMG [73]	https://github.com/HKUST-KnowComp/FMG	Python & C++
HeteRec [62]	https://github.com/mukulg17/HeteRec	R
DHNE [49]	https://github.com/tadpole/DHNE	Tensorflow
SHNE [66]	https://github.com/chuxuzhang/WSDM2019$_{SHNE}$	Pytorch
NSHE [74]	https://github.com/Andy-Border/NSHE	Pytorch
PAHNE [5]	https://github.com/chentingpc/GuidedHeteEmbedding	C++
Camel [64]	https://github.com/chuxuzhang/WWW2018$_{camel}$	Tensorflow
TaPEm [36]	https://github.com/pcy1302/TapEM	Python
HeGAN [21]	https://github.com/librahu/HeGAN	Tensorflow
DyHNE [56]	https://github.com/rootlu/DyHNE	Python & Matlab

- ArnetMiner (AMiner) [48]. In the early days, it was an academic network used for data mining. Now it becomes to a comprehensive academic system that provides a variety of academic resources. The address is https://www.aminer.cn/.
- Open Academic Society (OAS). It is an open and expanding knowledge graph for research and education, contributed by Microsoft Research and AMiner. It publishes Open Academic Graph (OAG), which unifies two billion-scale academic graphs. The address is https://www.openacademic.ai/.
- HG Resources. It is a website focusing on heterogeneous graphs, which collects a series of papers on HG and divides them into different categories, including classficiation, clustering and embedding. Code and datasets of the popular methods are also provided. The address is http://shichuan.org/.

2.3.3 Available Tools

Open-source platforms and toolkits can help researchers build the workflow of graph representation quickly and easily. Generally, there are many toolkits designed for homogeneous graph. For example, OpenNE[8] and CogDL[9]. However, the toolkits and platforms for heterogeneous graph are rarely mentioned. To bring this gap, we summary the popular toolkits and platforms that are suitable for heterogeneous graph.

- AliGraph. It is an industrial-grade machine learning platform for graph data, supporting the calculation of hundreds of millions of nodes and edges. Besides, it considers the characteristics of real-world industrial graph data, i.e., large-scale, heterogeneous, attributed and dynamic, and makes special optimizations. One instance can be found in https://www.aliyun.com/product/bigdata/product.
- Deep Graph Library (DGL). It is an open-source deep learning platform for graph data, which designs its own data structures and implements many popular methods. Specifically, it provides independent Application Programming Interfaces (APIs) for homogeneous graph, heterogeneous graph and knowledge graph. One instance can be found in https://www.dgl.ai/.
- Pytorch Geometric. It is a geometric deep learning extension library for pytorch. Specifically, it focuses on the methods for deep learning on graphs and other irregular structures. Same as DGL, it also has its own data structures and operators. One instance can be found in https://pytorch-geometric.readthedocs.io/en/latest/.
- OpenHINE. It is an open-source toolkit for HGR, which implements many popular HGR methods with a unified data interface. One instance can be found in https://github.com/BUPT-GAMMA/OpenHINE.

References

1. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: CIKM, pp. 891–900. ACM (2015)
2. Cen, Y., Zou, X., Zhang, J., Yang, H., Zhou, J., Tang, J.: Representation learning for attributed multiplex heterogeneous network. In: KDD. ACM (2019)
3. Chang, S., Han, W., Tang, J., Qi, G.J., Aggarwal, C.C., Huang, T.S.: Heterogeneous network embedding via deep architectures. In: KDD, pp. 119–128. ACM (2015)
4. Chen, H., Yin, H., Wang, W., Wang, H., Nguyen, Q.V.H., Li, X.: Pme: projected metric embedding on heterogeneous networks for link prediction. In: KDD, pp. 1177–1186. ACM (2018)
5. Chen, T., Sun, Y.: Task-guided and path-augmented heterogeneous network embedding for author identification. In: WSDM, pp. 295–304. ACM (2017)
6. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
7. Cui, P., Wang, X., Pei, J., Zhu, W.: A survey on network embedding. IEEE Transactions on Knowledge and Data Engineering **31**(5), 833–852 (2018)

[8] https://github.com/thunlp/OpenNE

[9] https://github.com/THUDM/cogdl

8. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: KDD, pp. 135–144. ACM (2017)
9. Fan, S., Zhu, J., Han, X., Shi, C., Hu, L., Ma, B., Li, Y.: Metapath-guided heterogeneous graph neural network for intent recommendation. In: KDD, pp. 2478–2486 (2019)
10. Fan, Y., Zhang, Y., Hou, S., Chen, L., Ye, Y., Shi, C., Zhao, L., Xu, S.: idev: Enhancing social coding security by cross-platform user identification between github and stack overflow. In: IJCAI, pp. 2272–2278 (2019)
11. Fu, T.y., Lee, W.C., Lei, Z.: Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In: CIKM, pp. 1797–1806. ACM (2017)
12. Fu, X., Zhang, J., Meng, Z., King, I.: Magnn: Metapath aggregated graph neural network forheterogeneous graph embedding. WWW (2020)
13. Fu, Y., Xiong, Y., Yu, P.S., Tao, T., Zhu, Y.: Metapath enhanced graph attention encoder for hins representation learning. In: BigData, pp. 1103–1110. IEEE (2019)
14. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: KDD, pp. 855–864. ACM (2016)
15. Gui, H., Liu, J., Tao, F., Jiang, M., Norick, B., Han, J.: Large-scale embedding learning in heterogeneous event data. In: ICDM, pp. 907–912. IEEE (2016)
16. Han, X., Shi, C., Wang, S., Yu, P.S., Song, L.: Aspect-level deep collaborative filtering via heterogeneous information networks. In: IJCAI, pp. 3393–3399 (2018)
17. He, Y., Song, Y., Li, J., Ji, C., Peng, J., Peng, H.: Hetespaceywalk: a heterogeneous spacey random walk for heterogeneous information network embedding. In: CIKM, pp. 639–648. ACM (2019)
18. Helgason, S.: Differential geometry, Lie groups, and symmetric spaces. Academic press (1979)
19. Hong, H., Guo, H., Lin, Y., Yang, X., Li, Z., Ye, J.: An attention-based graph neural network for heterogeneous structural learning. AAAI (2020)
20. Hou, S., Fan, Y., Zhang, Y., Ye, Y., Lei, J., Wan, W., Wang, J., Xiong, Q., Shao, F.: αcyber: Enhancing robustness of android malware detection system against adversarial attacks on heterogeneous graph based model. In: CIKM, pp. 609–618 (2019)
21. Hu, B., Fang, Y., Shi, C.: Adversarial learning on heterogeneous information networks. In: KDD, pp. 120–129. ACM (2019)
22. Hu, B., Shi, C., Zhao, W.X., Yu, P.S.: Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In: KDD, pp. 1531–1540. ACM (2018)
23. Hu, L., Li, C., Shi, C., Yang, C., Shao, C.: Graph neural news recommendation with long-term and short-term interest modeling. Information Processing & Management **57**(2), 102,142 (2020)
24. Hu, L., Xu, S., Li, C., Yang, C., Shi, C., Duan, N., Xie, X., Zhou, M.: Graph neural news recommendation with unsupervised preference disentanglement. In: ACL (2020)
25. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. WWW (2020)
26. Hussein, R., Yang, D., Cudré-Mauroux, P.: Are meta-paths necessary?: Revisiting heterogeneous graph embeddings. In: CIKM, pp. 437–446 (2018)
27. Jamali, M., Lakshmanan, L.V.S.: Heteromf: recommendation in heterogeneous information networks using context dependent factor models. In: WWW, pp. 643–654. International World Wide Web Conferences Steering Committee / ACM (2013)
28. Jeh, G., Widom, J.: Scaling personalized web search. In: WWW, pp. 271–279 (2003)
29. Lee, S., Park, C., Yu, H.: Bhin2vec: Balancing the type of relation in heterogeneous information network. In: CIKM, pp. 619–628 (2019)
30. Li, C., Hu, L., Shi, C., Song, G., Lu, Y.: Sequence-aware heterogeneous graph neural collaborative filtering. In: SDM (2021)
31. Liu, Z., Zheng, V.W., Zhao, Z., Li, Z., Yang, H., Wu, M., Ying, J.: Interactive paths embedding for semantic proximity search on heterogeneous graphs. In: KDD, pp. 1860–1869. ACM (2018)
32. Liu, Z., Zheng, V.W., Zhao, Z., Yang, H., Chang, K.C., Wu, M., Ying, J.: Subgraph-augmented path embedding for semantic user search on heterogeneous social network. In: WWW, pp. 1613–1622. ACM (2018)

33. Liu, Z., Zheng, V.W., Zhao, Z., Zhu, F., Chang, K.C.C., Wu, M., Ying, J.: Distance-aware dag embedding for proximity search on heterogeneous graphs. In: AAAI (2018)
34. Lu, Y., Shi, C., Hu, L., Liu, Z.: Relation structure-aware heterogeneous information network embedding. In: AAAI (2019)
35. Matsuno, R., Murata, T.: MELL: effective embedding method for multiplex networks. In: WWW, pp. 1261–1268. ACM (2018)
36. Park, C., Kim, D., Zhu, Q., Han, J., Yu, H.: Task-guided pair embedding in heterogeneous network. In: CIKM, pp. 489–498 (2019)
37. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: KDD, pp. 701–710 (2014)
38. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference, pp. 593–607. Springer (2018)
39. Shi, C., Hu, B., Zhao, W.X., Yu, P.S.: Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering **31**(2), 357–370 (2018)
40. Shi, C., Kong, X., Huang, Y., Yu, P.S., Wu, B.: Hetesim: A general framework for relevance measure in heterogeneous networks. IEEE Trans. Knowl. Data Eng. **26**(10), 2479–2492 (2014)
41. Shi, C., Li, Y., Zhang, J., Sun, Y., Yu, P.S.: A survey of heterogeneous information network analysis. IEEE Trans. Knowl. Data Eng. **29**(1), 17–37 (2017)
42. Shi, C., Zhang, Z., Luo, P., Yu, P.S., Yue, Y., Wu, B.: Semantic path based personalized recommendation on weighted heterogeneous information networks. In: CIKM, pp. 453–462. ACM (2015)
43. Shi, Y., Gui, H., Zhu, Q., Kaplan, L., Han, J.: Aspem: Embedding learning by aspects in heterogeneous information networks. In: SDM, pp. 144–152. SIAM (2018)
44. Shi, Y., Zhu, Q., Guo, F., Zhang, C., Han, J.: Easing embedding learning by comprehensive transcription of heterogeneous information networks. In: KDD, pp. 2190–2199. ACM (2018)
45. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the VLDB Endowment **4**(11), 992–1003 (2011)
46. Tang, J., Qu, M., Mei, Q.: Pte: Predictive text embedding through large-scale heterogeneous text networks. In: KDD, pp. 1165–1174. ACM (2015)
47. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: WWW, pp. 1067–1077 (2015)
48. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: KDD, pp. 990–998. ACM (2008)
49. Tu, K., Cui, P., Wang, X., Wang, F., Zhu, W.: Structural deep embedding for hyper-networks. In: AAAI (2018)
50. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: KDD, pp. 1225–1234 (2016)
51. Wang, H., Wang, J., Wang, J., Zhao, M., Zhang, W., Zhang, F., Xie, X., Guo, M.: Graphgan: Graph representation learning with generative adversarial nets. In: AAAI (2018)
52. Wang, R., Shi, C., Zhao, T., Wang, X., Ye, Y.: Heterogeneous information network embedding with adversarial disentangler. IEEE Transactions on Knowledge and Data Engineering (2021)
53. Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., Yang, S.: Community preserving network embedding. In: AAAI (2017)
54. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: WWW, pp. 2022–2032. ACM (2019)
55. Wang, X., Liu, N., Han, H., Shi, C.: Self-supervised heterogeneous graph neural network with co-contrastive learning. In: KDD (2021)
56. Wang, X., Lu, Y., Shi, C., Wang, R., Cui, P., Mou, S.: Dynamic heterogeneous information network embedding with meta-path based proximity. IEEE Transactions on Knowledge and Data Engineering (2020)

57. Wang, X., Zhang, Y., Shi, C.: Hyperbolic heterogeneous information network embedding. In: AAAI (2019)
58. Wang, Z., Liu, H., Du, Y., Wu, Z., Zhang, X.: Unified embedding model over heterogeneous information network for personalized recommendation. In: IJCAI, pp. 3813–3819. AAAI Press (2019)
59. Xu, L., Wei, X., Cao, J., Yu, P.S.: Embedding of embedding (eoe): Joint embedding for coupled heterogeneous networks. In: WSDM, pp. 741–749. ACM (2017)
60. Xu, Y., Zhu, Y., Shen, Y., Yu, J.: Learning shared vertex representation in heterogeneous graphs with convolutional networks for recommendation. In: IJCAI, pp. 4620–4626 (2019)
61. Xue, H., Yang, L., Jiang, W., Wei, Y., Hu, Y., Lin, Y.: Modeling dynamic heterogeneous network for link prediction using hierarchical attention with temporal rnn. arXiv preprint arXiv:2004.01024 (2020)
62. Yu, X., Ren, X., Sun, Y., Sturt, B., Khandelwal, U., Gu, Q., Norick, B., Han, J.: Recommendation in heterogeneous information networks with implicit user feedback. In: RecSys, pp. 347–350. ACM (2013)
63. Yun, S., Jeong, M., Kim, R., Kang, J., Kim, H.J.: Graph transformer networks. In: NIPS, pp. 11,960–11,970 (2019)
64. Zhang, C., Huang, C., Yu, L., Zhang, X., Chawla, N.V.: Camel: Content-aware and meta-path augmented metric learning for author identification. In: WWW, pp. 709–718 (2018)
65. Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V.: Heterogeneous graph neural network. In: KDD, pp. 793–803. ACM (2019)
66. Zhang, C., Swami, A., Chawla, N.V.: Shne: Representation learning for semantic-associated heterogeneous networks. In: WSDM, pp. 690–698 (2019)
67. Zhang, D., Yin, J., Zhu, X., Zhang, C.: Metagraph2vec: complex semantic path augmented heterogeneous network embedding. In: PAKDD, pp. 196–208. Springer (2018)
68. Zhang, H., Qiu, L., Yi, L., Song, Y.: Scalable multiplex network embedding. In: IJCAI, vol. 18, pp. 3082–3088 (2018)
69. Zhang, Y., Fan, Y., Song, W., Hou, S., Ye, Y., Li, X., Zhao, L., Shi, C., Wang, J., Xiong, Q.: Your style your identity: Leveraging writing and photography styles for drug trafficker identification in darknet markets over attributed heterogeneous information network. In: WWW, pp. 3448–3454 (2019)
70. Zhang, Y., Fan, Y., Ye, Y., Zhao, L., Shi, C.: Key player identification in underground forums over attributed heterogeneous information network embedding framework. In: CIKM, pp. 549–558 (2019)
71. Zhang, Z., Cui, P., Wang, X., Pei, J., Yao, X., Zhu, W.: Arbitrary-order proximity preserved network embedding. In: KDD, pp. 2778–2786. ACM (2018)
72. Zhao, H., Yao, Q., Li, J., Song, Y., Lee, D.L.: Meta-graph based recommendation fusion over heterogeneous information networks. In: KDD, pp. 635–644. ACM (2017)
73. Zhao, H., Zhou, Y., Song, Y., Lee, D.L.: Motif enhanced recommendation over heterogeneous information network. In: CIKM, pp. 2189–2192 (2019)
74. Zhao, J., Wang, X., Shi, C., Liu, Z., Ye, Y.: Network schema preserving heterogeneous information network embedding. In: IJCAI (2020)
75. Zhao, J., Zhou, Z., Guan, Z., Zhao, W., Ning, W., Qiu, G., He, X.: Intentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation. In: KDD, pp. 2347–2357 (2019)
76. Zhao, K., Bai, T., Wu, B., Wang, B., Zhang, Y., Yang, Y., Nie, J.: Deep adversarial completion for sparse heterogeneous information network embedding. In: WWW, pp. 508–518. ACM / IW3C2 (2020)
77. Zhu, S., Zhou, C., Pan, S., Zhu, X., Wang, B.: Relation structure-aware heterogeneous graph neural network. In: IEEE International Conference On Data Mining (2019)