Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions

CHEN GAO*[1], YU ZHENG*[1], NIAN LI[1], YINFENG LI[1], YINGRONG QIN[1], JINGHUA PIAO[1], YUHAN QUAN[1], JIANXIN CHANG[1], DEPENG JIN[1], XIANGNAN HE[2], YONG LI*[1], [1]Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University and [2]School of Information Science and Technology, University of Science and Technology of China

Recommender system is one of the most important information services on today's Internet. Recently, graph neural networks have become the new state-of-the-art approach of recommender systems. In this survey, we conduct a comprehensive review of the literature in graph neural network-based recommender systems. We first introduce the background and the history of the development of both recommender systems and graph neural networks. For recommender systems, in general, there are four aspects for categorizing existing works: stage, scenario, objective, and application. For graph neural networks, the existing methods consist of two categories, spectral models and spatial ones. We then discuss the motivation of applying graph neural networks into recommender systems, mainly consisting of the high-order connectivity, the structural property of data, and the enhanced supervision signal. We then systematically analyze the challenges in graph construction, embedding propagation/aggregation, model optimization, and computation efficiency. Afterward and primarily, we provide a comprehensive overview of a multitude of existing works of graph neural network-based recommender systems, following the taxonomy above. Finally, we raise discussions on the open problems and promising future directions of this area. We summarize the representative papers along with their codes repositories in https://github.com/tsinghua-fib-lab/GNN-Recommender-Systems.

1 INTRODUCTION

Recommender system, is a kind of filtering system of which the goal is to present personalized information to users, which improves the user experience and promotes business profit. As one of

---

*Yong Li is the *Corresponding Author*. The first two authors contributed equally to this paper.

Author's address: Chen Gao*[1], Yu Zheng*[1], Nian Li[1], Yinfeng Li[1], Yingrong Qin[1], Jinghua Piao[1], Yuhan Quan[1], Jianxin Chang[1], Depeng Jin[1], Xiangnan He[2], Yong Li*[1], [1]Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, [2]School of Information Science and Technology, University of Science and Technology of China, chgao96@gmail.com,{y-zheng19,linian21,liyf19,qyr16,pojh19,quanyh19, changjx18}@mails.tsinghua.edu.cn,xiangnanhe@gmail.com,{jindp,liyong07}@tsinghua.edu.cn.

---

the typical applications of machine learning driven by the real world, it is an extremely hot topic in both industrial and academia nowadays.

To recap the history of recommender systems, it can be generally divided into three stages, shallow models [74, 125, 126], neural models [26, 48, 56], and GNN-based models [55, 153, 188]. The earliest recommendation models capture the collaborative filtering (CF) effect by directly calculating the similarity of interactions. Then model-based CF methods, such as matrix factorization (MF) [74] or factorization machine[125], were proposed to approach recommendation as a representation learning problem. However, these methods are faced with critical challenges such as complex user behaviors or data input. To address it, neural network-based models [26, 48, 56] are proposed. For example, neural collaborative filtering (NCF) was developed to extend the inner product in MF with multi-layer perceptrons (MLP) to improve its capacity. Similarly, deep factorization machine (DeepFM) [48] combined the shallow model factorization machine (FM) [125] with MLP. However, these methods are still highly limited since their paradigms of prediction and training ignore the high-order structural information in observed data. For example, the optimization goal of NCF is to predict user-item interaction, and the training samples include observed positive user-item interactions and unobserved negative user-item interactions. It means that during the parameter updating for a specific user, only the items interacted by him/her are involved.

Recently, the advances of graph neural networks provide a strong fundamental and opportunity to address the above issues in recommender systems. Specifically, graph neural networks adopt embedding propagation to aggregate neighborhood embedding iteratively. By stacking the propagation layers, each node can access high-order neighbors' information, rather than only the first-order neighbors as the traditional methods do. With its advantages to handle the structural data and to explore structural information, GNN-based methods have become the new state-of-the-art approaches in recommender systems.

To well apply graph neural networks into recommender systems, there are some critical challenges required to be addressed. First, the data input of recommender system should be carefully and properly constructed to graph, with nodes representing elements and edges representing the relations. Second, for the specific task, the component in the graph neural network should be adaptively designed, including how to propagate and aggregate, in which existing works have explored various choices with different advantages and disadvantages. Third, the optimization of the GNN-based model, including the optimization goal, loss function, data sampling, etc., should be consistent with the task requirement. Last, since recommender systems have strict limitations on the computation cost, and also due to GNNs' embedding propagation operations introduce a number of computations, the efficient deployment of graph neural networks in recommender systems is another critical challenge.

In this paper, we aim to provide a systematic and comprehensive review of the research effort, especially on how they improve recommendation with graph neural networks and address the corresponding challenges. To fulfill a clear understanding, we categorize researches of recommender systems from four perspectives, stage, scenario, objectives, and applications. We summarize the representative papers along with their codes repositories in https://github.com/tsinghua-fib-lab/GNN-Recommender-Systems.

It is worth mentioning that there is one existing survey [168] of graph neural network-based recommender system. However, it is limited due to the following reasons. First, it does not provide extensive taxonomy of recommender systems. Specifically, it roughly divides recommender systems into non-sequential recommendation and sequential recommendation, however, which is not so reasonable. In fact, the sequential recommendation is only one specific recommendation scenario with a special setting of input and output, as pointed out by this survey. Second, it does not provide adequate explanations of the motivations and reasons that the existing works leverage graph
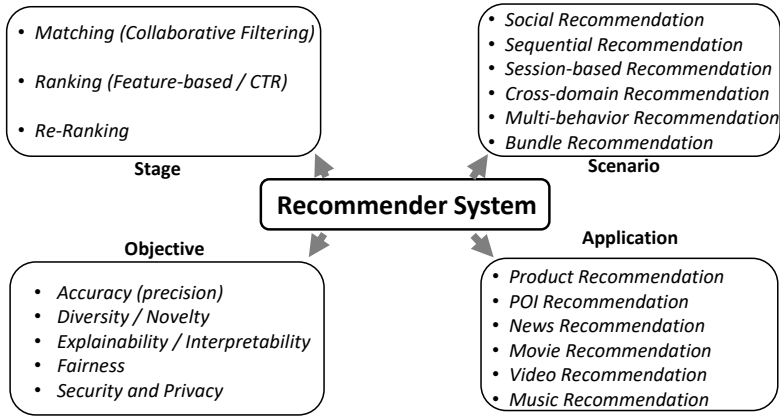
Fig. 1. An illustration of typical recommender systems (stages, scenarios, objectives, and applications)

neural networks for recommender systems. In this survey, by contrast, we provide a comprehensive understanding of why GNN can be and should be used in recommender system, which can help readers to understand the position and value of this new research field. Third, it does not explain the critical challenges of applying graph neural networks to recommendation and how to address them, which are fully discussed in this survey. Last, since this area is increasingly popular, our survey also introduces a lot of recently published papers not covered by [168].

The structure of this survey is organized as follows. We first introduce the background of recommender systems, from four kinds of perspectives (stage, scenario, objective, application), and the background of graph neural networks, in Section 2. We then discuss the challenges of applying graph neural networks to recommender systems from four aspects, in Section 3. Then we elaborate on the representative methods of graph nerual network-based recommendation in Section 4 by following the taxonomy in the above section. We discuss the most critical open problems in this area and provide ideas of the future directions in Section 5 and conclude this survey in Section 6.

2 BACKGROUND

2.1 Recommender Systems

2.1.1 Overview . In this section, we present the background of recommender systems from four perspectives: stages, scenarios, objectives, and applications. Specifically, in industrial applications, due to the real-world requirements on system engineering, the recommender systems are always split into three stages, *matching*, *ranking*, and *re-ranking*, forming a standard pipeline. Each stage has different characteristics on data input, output, model design, etc. Besides the standard stages, there are many specific recommendation scenarios with a special definition. For example, in the last twenty years, social recommendation has been attracting attention, defined as improving recommender system based on social relations. Last, different recommender systems have different objectives, of which accuracy is always the most important one as it directly determines the system's utility. Recently, recommender systems have been assigned other requirements such as recommending diversified items to avoid boring user experience, making sure the system treats all users fairly, protecting user privacy from attack, etc. As for the applications, GNN models can

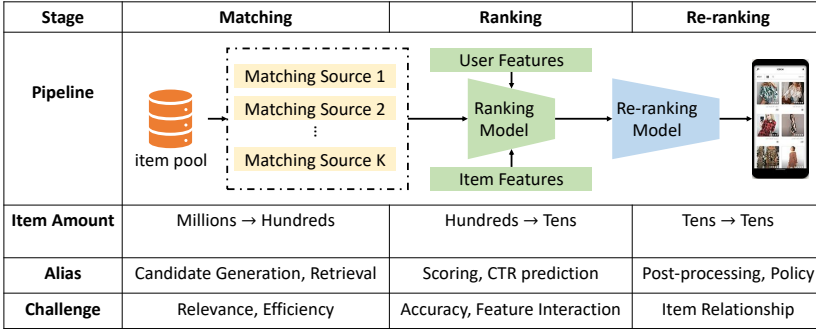| Stage	Matching	Ranking	Re-ranking
Pipeline	item pool → Matching Source 1 / Matching Source 2 / Matching Source K	User Features → Ranking Model ← Item Features	Re-ranking Model
Item Amount	Millions → Hundreds	Hundreds → Tens	Tens → Tens
Alias	Candidate Generation, Retrieval	Scoring, CTR prediction	Post-processing, Policy
Challenge	Relevance, Efficiency	Accuracy, Feature Interaction	Item Relationship

Fig. 2. The typical pipeline of recommender systems.

be widely deployed in e-commerce recommendation, point-of-interest recommendation, news recommendation, movie recommendation, music recommendation, etc.

2.1.2 Stages . The item pool, *i.e.* all the items available for recommender systems, is usually large and can include millions of items. Thus, common recommender systems follow a *multi-stage* architecture, filtering items stage by stage from the large-scale item pool to the final recommendations exposed to users, tens of items. [29, 158]. Generally, a modern recommender system is composed of the following three stages.

- **Matching.** This first stage generates hundreds of candidate items from the extremely large item pool (million-level or even billion-level), which significantly reduces the scale. Considering the large scale of data input in this stage, and due to the strict latency restrictions of online serving, complicated algorithms cannot be adopted, such as very deep neural networks [29, 70]. In other words, models in this stage are usually concise. That is, the core task of this stage is to retrieve potentially relevant items with high efficiency and attain coarse-grained modeling of user interests. It is worth noting that a recommender system in the real world usually contains multiple matching channels with multiple models, such as embedding matching, geographical matching, popularity matching, social matching, etc.
- **Ranking.** After the matching stage, multiple sources of candidate items from different channels are merged into one list and then scored by a single ranking model. Specifically, the ranking model ranks these items according to the scores, and the top dozens of items are selected. Since the amount of input items in this stage is relatively small, the system can afford much more complicated algorithms to achieve higher recommendation accuracy [69, 89, 133]. For example, rich features including user profiles and item attributes can be taken into consideration, and advanced techniques such as self-attention [69] can be utilized. Since a lot of features are involved, the key challenge in this stage is to design appropriate models for capturing complicated feature interactions.
- **Re-ranking.** Although the obtained item list after the ranking stage is optimized with respect to relevance, it may not meet other important requirements, such as freshness, diversity, fairness, etc. [120] Therefore, a re-ranking stage is necessary, which usually removes certain items or changes the order of the list in order to fulfill additional criteria and also satisfy business needs. The main concern in this stage is to consider multiple relationships among the top-scored items [5, 217]. For example, similar or substitutable items can lead to information redundancy when they are displayed closely in the recommendations.
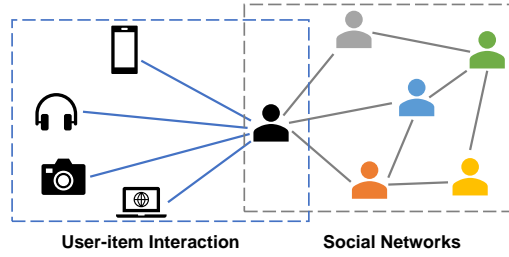
Fig. 3. An illustration of social recommendation. User interactions are affected by both their own preferences and the social factor.
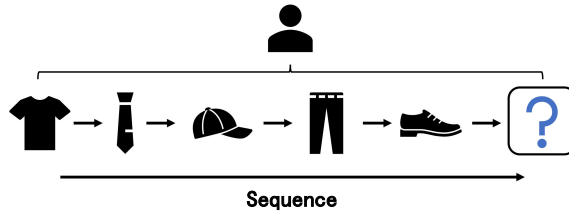

Fig. 4. An illustration of sequential recommendation. Given a user's historical sequence, recommender system aims to predict the next item.

Fig. 2 illustrates the typical pipeline of a recommender system, as well as the comparisons of the three stages above.

2.1.3 Scenarios. In the following, we will elaborate on the different scenarios of recommender systems, including social recommendation, sequential recommendation, session recommendation, bundle recommendation, cross-domain recommendation, and multi-behavior recommendation.

- **Social Recommendation.** In the past few years, social platforms have dramatically changed users' daily life. With the ability to interact with other users, individual behaviors are driven by both personal and social factors. Specifically, users' behavior may be influenced by what their friends might do or think, which is known as *social influence* [28]. For example, users in WeChat* Video platform may *like* some videos only because of their WeChat friends' *like* behaviors. At the same time, *social homophily* is another popular phenomenon in many social platforms, *i.e.*, people tend to build social relations with others who have similar preferences with them [107]. Taking social e-commerce as an example, users from a common family possibly share similar product preferences, such as food, clothes, daily necessities, and so on. Hence, social relations are often integrated into recommender systems to enhance the final performance, which is called social recommendation. Fig. 3 illustrates the data input of social recommendation, of which user interactions are determined by both his/her own preferences and social factors (social influence and social homophily).
- **Sequential Recommendation.** In recommender systems, users will produce a large number of interaction behaviors over time. The sequential recommendation method extracts information from these behavioral sequences and predicts the user's next interaction item, as shown in Fig. 4. To symbolize this problem, for the sequence of items $\{x_1, x_2, ..., x_n\}$ that the user has

---

*http://wechat.com/

Fig. 5. An illustration of session-based recommendation. Given a anonymous short session, recommender system aims to predict the next item.
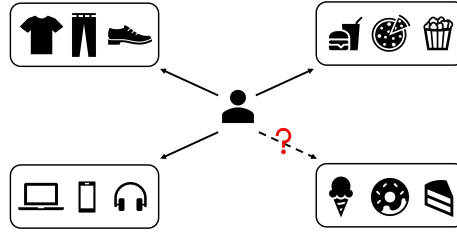

Fig. 6. An illustration of bundle recommendation.

interacted with in order, the goal of the system is to predict the next item x_{n+1} that the user will interact with. In recommender systems, the user's historical behaviors play an important role in modeling the user's interest. Many commonly used recommendation methods like collaborative filtering [56] train the model by taking each user behavior as a sample. They directly model the user's preference on a single item, but sequential recommendation uses the user's historical behavior sequence to learn timestamp-aware sequential patterns to recommend the next item that the user may be interested in. In sequential recommendation, there are two main challenges. First of all, for each sample, *i.e.*, each sequence, the user's interest needs to be extracted from the sequence to predict the next item. Especially when the sequence length increases, it is very challenging to simultaneously model the short-term, long-term, and dynamic interests of users. Secondly, in addition to modeling within a sequence, since items may occur in multiple sequences or users have multiple sequences, collaborative signals between different sequences require to be captured for better representation learning.

- **Session-based Recommendation.** In many real-world scenarios, such as some small retailers and mobile stream media (*e.g.* YouTube and Tiktok), it is impossible or not necessary to track the user-id's behaviors over a long period of time due to limited storage resources. In other words, user profiles and long-term historical interactions are unavailable, and only the short session data from anonymous users are provided. Hence, conventional recommendation methods (*e.g.*, collaborative filtering) may perform poorly in this scenarios. This motivates the problem of session-based recommendation (SBR), which aims at predicting the next item with a given anonymous behavioral session data as shown in Fig. 5. Distinct from the sequential recommendation, the subsequent sessions of the same user are handled independently in SBR, since the behavior of users in each session only shows session-based traits [57].

- **Bundle Recommendation.** The existing recommender systems mainly focus on recommending independent items to users. The bundle is a collection of items, which is an important marketing strategy for product promotion. Bundle recommendation aims to recommend a combination of items for users to consume [11, 22, 97, 118]. Bundle recommendation is very common nowadays in online platforms, e.g., the music playlists on Spotify, the pinboards on Pinterest, the computer
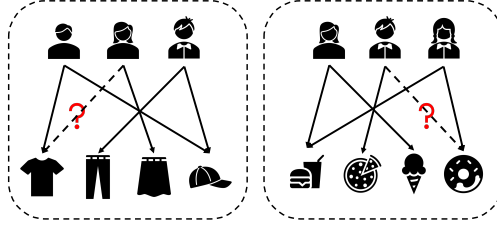
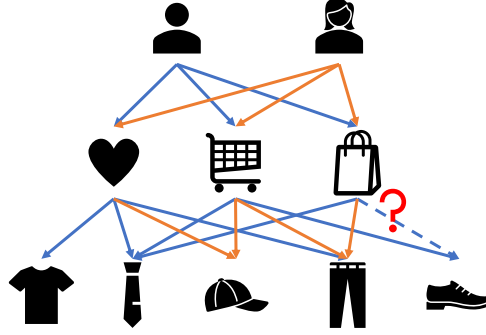Fig. 7. An illustration of cross-domain recommendation.

Fig. 8. An illustration of multi-behavior recommendation.

sets on Amazon, and the furniture suites on IKEA. It is worth mentioning that bundle recommendations are also used to solve interesting and meaningful problems such as fashion outfits [83] and drug packages [209].

- **Cross-Domain Recommendation.** As an increasing number of users interact with multi-modal information across multiple domains, cross-domain recommendation (CDR) has been proven to be a promising method to alleviate cold start and data sparsity problems [41, 43, 60, 68, 104, 112, 203]. CDR methods can be roughly divided into two categories, single-target CDR (STCDR) and dual-target CDR (DTCDR) [30]. CDR methods transfer the information from the source domain to the target domain in one direction; DTCDR emphasizes the mutual utilization of information from both source domain and target domain, which can be extended to multiple-target CDR (MTCDR). Since utilizing information from multiple domains can improve performance, cross-domain recommendation has become an important scenario in recommender systems.

- **Multi-behavior Recommendation.** Users interact with recommender systems under multiple types of behaviors instead of only one type of behavior. For example, when the user clicks on the video, he/she may also perform behaviors such as collecting or comment. In an e-commerce website, users often click, add into shopping carts, share, or collect a product before purchasing it, as shown in Fig. 8. Although the ultimate goal of the recommender system is to recommend products that users will purchase, the purchase behavior is very sparse compared to the user's click, sharing, and other behaviors. To symbolize this problem, For each user u and item v, suppose there are K different types of behaviors $\{y_1, y_2, ..., y_K\}$. For the i-th behavior, if the user has a observed behavior, then $y_i = 1$, otherwise $y_i = 1$. The goal of the recommender system is to improve the prediction accuracy of a certain type of target behavior y_t.
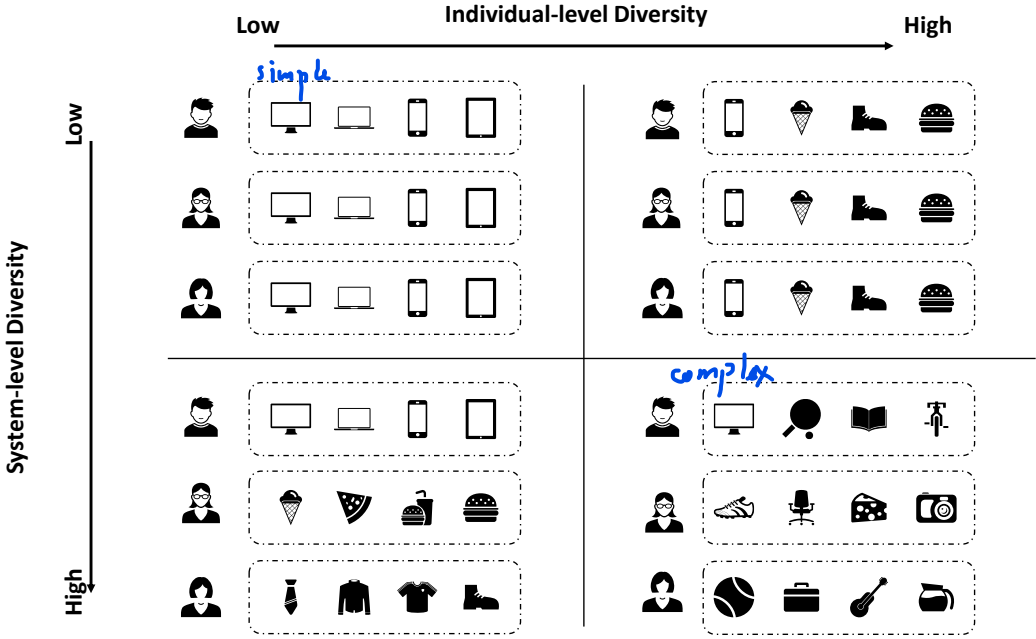
Fig. 9. Illustration of individual-level diversity and system-level diversity.

For multi-behavior recommendation, generally speaking, there are two main challenges. Firstly, different behaviors have different influences on the target behavior. Some behaviors may be strong signals, and some may be weak signals [173, 174]. At the same time, this influence is different for each user. It's challenging to model the influence of these different behaviors on the target behavior accurately. Secondly, it is challenging to learn comprehensive representations from different types of behaviors for items. Different behaviors reflect users' different preferences for items; in other words, different behaviors have different meanings. In order to obtain a better representation, the meaning of different behaviors needs to be integrated into the representation learning.

2.1.4 Objectives. The most important objective of recommender systems, of course, is accuracy. In the following, we elaborate on the three other important objectives, including diversity, explainability, and fairness.

- **Diversity.** Two types of diversity are usually considered in recommender systems, namely individual-level diversity and system-level diversity. Specifically, individual-level diversity is an important objective which measures the dissimilarity of the recommended items for each user since repeated similar items make users reluctant to explore with the system. In other words, individual-level diversity reflects how many topics the recommendation list covers and how balanced is the recommended items distributed on different topics. Here topics depend on the recommendation task, *e.g.* topics can be different product categories for e-commerce recommendation and different genres for music recommendation [205]. Regarding system-level diversity, it compares the recommendation results of different users and expects them to be dissimilar with each other. In other words, low system-level diversity means always recommending popular items to all the users while ignoring long-tail items. Therefore, system-level diversity is

sometimes called *long-tail* recommendation. Fig. 9 briefly illustrates the two types of diversity and their differences. For both individual-level and system-level diversity, there are two main challenges. First, the signal strength of different items varies greatly. For each user, there exist dominant topics and disadvantaged topics, *e.g.* a user's interaction records with electronics might be much more frequent than with clothes. Similarly, the signal strength of long-tail items is also far weaker than popular items. Therefore, it is challenging to recommend relevant content with such weak supervision from either disadvantaged topics or long-tail items for individual-level and system-level diversity, respectively. Second, diversity may sometimes contradict recommendation accuracy, resulting in the accuracy-diversity dilemma; thus it is challenging to balance between the two aspects.

- **Explainability.** As current recommender systems mostly adopt a deep learning paradigm, they can arouse urgent needs on the explainability of recommendation [201]. The focus of explainable recommender systems is not only to produce accurate recommendation results but to generate persuasive explanations for how and why the item is recommended to a specific user [201]. Increasing the explainability of recommender systems can enhance users' perceived transparency [131], persuasiveness [142] and trustworthiness [75], and facilitate practitioners to debug and refine the system [201]. This survey mainly focuses on improving explainability with machine learning techniques. Specifically, past research adopts two different approaches [201]: one makes efforts to design intrinsic explainable models, ensuring the explainability of recommendation results by designing models with transparent logic (rather than merely "black box"), e.g., Explicit Factor Models [202], Hidden Factor and Topic Model [106], and TriRank [54]. The others compromise slightly: they design post-hoc separate models to explain the results generated by "black box" recommender systems, e.g., Explanation Mining [119]. Currently, there are two challenges. First, representing explainable information requires graph-structural item attributes, which are difficult to model without the power of GNN. Second, reasoning recommendations depend on external knowledge in the knowledge graph, which also poses challenges on the task.

- **Fairness.** As a typical data-driven system, recommender systems could be biased by data and the algorithm, arousing increasing concerns on the fairness [4, 86, 108]. Specifically, according to the involved stakeholders, fairness in recommender systems can be divided into two categories [4, 86, 108]: user fairness, which attempts to ensure no algorithmic bias among specific users or demographic groups [9, 78, 84], and item fairness, which indicates fair exposures of different items, or no popularity bias among different items [2, 3, 86, 108]. Here, we focus on the user fairness, and leave item fairness in the section of diversity for their close connection in terms of interpretations and solutions [2, 105]. Specifically, researchers adopt two methods to enhance fairness: One is directly debiasing recommendation results in the training process [9, 216], while the other endeavors to rank items to alleviate the unfairness in a post-processing method [46, 130]. Indeed, increasing evidence shows that the utilization of graph data, e.g., user-user, could intensify concerns on fairness [31, 124, 162]. Thus, it is challenging to debias unfairness in recommendation in the context of rich graph data. Furthermore, it is even harder to boost user fairness in recommendation from the perspective of graph.

2.1.5 Applications. Recommender systems widely existing in today's information services, with various kinds of applications, of which the representative ones are as follows.

Product recommendation, also known as E-commerce recommendation, is one the most famous applications of recommender systems. For recommendation models in the e-commerce scenario, the business value is highly concerned. Therefore, it is crucial to handle multiple types of behaviors closely relevant to the platform profit, including adding to cart or purchase. Some works [102] propose to optimize the click-through rate and conversion rate at the same time. In addition, in

Table 1. Frequently used notations.

V/E	The set of graph nodes/edges
$\mathcal{U}/\mathcal{I}$	The set of users/items
$\mathcal{N}_i$	The neighborhood set of graph node i
$\mathbf{h}_i^l$	The embedding of graph node i in the l-th propagation layer
$\mathbf{A}$	The adjacency matrix of graph
$\mathbf{W}^l$	Learnable transformation matrix in the l-th propagation layer
$\delta(\cdot)$	Nonlinear activation function
$\parallel$	Concatenation operation
$\odot$	Hadamard product

e-commerce platform, products may have rich attributes, such as price [207], category [210], etc., based on which heterogeneous graphs can be constructed [99]. Representative benchmark datasets for product recommendation includes Amazon [†], Tmall [‡].

POI (Point-of-interest) recommendation, is also a popular application that aims to recommend new locations/point-of-interests for users' next visitation. In Point-of-interest recommendation, there are two important factors, spatial factor and temporal factor. The spatial factor refers to naturally-existed geographical attributes of POIs, *i.e.*, the geographic location. In addition, the users' visitations are also largely limited by their geographical activity areas since the user cannot visit POIs as easily as browsing/purchasing products on e-commerce websites. Besides, the temporal factor is also of great importance since users' visitation/check-in behaviors always form a sequence. This motivates the problem of next-POI or successive POI recommendation[37, 90, 179]. Representative benchmark datasets for news recommendation includes Yelp [§], Gowalla [¶], etc.

News recommendation helps users find preferred news, which is also another typical application. Different from other recommendation applications, news recommendation requires proper modeling of the texts of news. Therefore, methods of natural language processing can be combined with recommendation models for extracting news features better [114]. Besides, users are always interested in up-to-date news and may refuse out-of-date ones. Thus, it is also critical but challenging to fast and accurately filter news from the fast-changing candidate pools. As for news recommendation, MIND dataset [159] is the recently-released representative benchmark dataset.

Movie recommendation is one of the earliest recommender systems. Precisely, Netflix's movie recommendation competition [7] motivated many pioneer recommendation researches [7, 73]. The earlier setting of movie recommendation is to estimate the users' rating scores on movies, from one to five, which is named *explicit feedback*. Recently, the binary *implicit feedback* has become more popular setting [56, 126].

There are also other types of recommendation applications, such as video recommendation [32, 93], music recommendation [143, 156], job recommendation [117], food recommendation [111], etc.

2.2 Graph Neural Networks

With the rapid emergence of vast volumes of graph data, such as social networks, molecular structures, and knowledge graphs, a wave of graph neural network (GNN) studies has sprung up in recent years [8, 38, 40, 72, 144, 195]. The rise of GNN mainly originates from the advancement of

---

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

[‡]https://tianchi.aliyun.com/dataset/dataDetail?dataId=649

[§]https://www.yelp.com/dataset

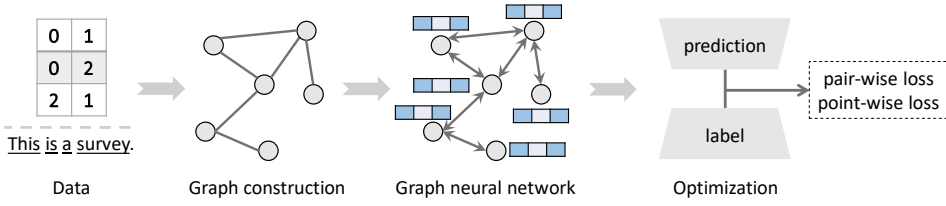[¶]https://snap.stanford.edu/data/loc-gowalla.html

Fig. 10. The overall procedure of implementing a GNN model: construct graphs from data (*e.g.*, table or text), design tailored GNN for generating representations, map representations to prediction results, and further define loss function with labels for optimization.

convolutional neural network (CNN) and graph representation learning (GRL) [171, 212]. When applied to regular Euclidean data such as images or texts, CNN is extremely effective in extracting localized features. However, for non-Euclidean data like graphs, CNN requires generalization to handle the situations where operation objects (*e.g.*, pixels in images or nodes on graphs) are non-fixed in size. In terms of GRL, it aims to generate low-dimensional vectors for graph nodes, edges, or subgraphs, which represent complex connection structures of graphs. For example, a pioneering work DeepWalk [121] learns node representations by using SkipGram [110] on a generated path with random walks on graphs. Combining CNN and GRL, various GNNs are developed to distill structural information and learn high-level representations. In later parts, we will introduce several general and primary stages for designing a GNN model to accomplish tasks on graphs, illustrated in Fig. 10. Specifically, section 2.2.1, 2.2.2 and 2.2.3 elaborate how to construct graphs, design specialized and effective graph neural networks and optimize models, respectively.

2.2.1 Graph Construction. We first utilize the unified formula to define a graph, *i.e.*, $\mathcal{G} = (V, E)$, where V and E denote the set of graph nodes and edges respectively, and each edge in E joins any number of nodes in V. In recent years, GNN-based models focus on designing specialized networks for the following three categories of graphs

- **Homogeneous graph**, of which each edge connects only two nodes, and there is only one type of nodes and edges.
- **Heterogeneous graph**, of which each edge connects only two nodes, and there are multiple types of nodes or edges.
- **Hypergraph**, of which each edge joins more than two nodes.

In many information services nowadays, relational data is naturally represented in the form of graphs. For example, implicit social media relationships can be considered as a unified graph, with nodes representing individuals and edges connecting people who follow each other. However, since non-structured data such as images and texts do not explicitly contain graphs, it is necessary to define nodes and edges manually for building graphs. Taking text data used in Natural Language Processsing (NLP) as an example, words/documents are described as nodes, and edges among them are constructed according to Term Frequency-Inverse Document Frequency (IF-ITF) [186]. Additionally, an emerging research direction in representation learning on graphs is knowledge graph (KG), which is a representative instance of the heterogeneous graph. KG integrates multiple data attributes and relationships, in which nodes and edges are redefined as entities and relations, respectively. Specifically, the entities in KG can cover a wide range of elements, including persons, movies, books, and so on. The relations are utilized to describe how entities associate with each other. For example, a movie can relate to persons (*e.g.*, actors or directors), countries, languages, *etc.* Except for ordinary graphs, the hypergraph is also explored recently to handle more complex

data (*e.g.*, beyond pairwise relations and multiple modals) flexibly [38], in which each edge can connect more than two nodes.

In summary, constructing graphs necessitates either pre-existing graph data or abstracting the concept of graph nodes and edges from non-structured data.

2.2.2 Network Design. Generally speaking, GNN models can be categorized into spectral and spatial models. Spectral models consider graphs as signals and process them with graph convolution in the spectral domain. Specifically, graph signals are first transformed into spectral domain by Fourier transform defined on graphs, then a filter is applied, at last the processed signals are transformed back to spatial domain [129]. The formulation of processing the graph signal $\mathbf{x}$ with the filter $\mathbf{g}$ is

$$\mathbf{g} \star \mathbf{x} = \mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{g}) \odot \mathcal{F}(\mathbf{x})\right), \tag{1}$$

where $\mathcal{F}$ denotes the graph Fourier transform.

In contrast, spatial models conduct the convolution on graph structures directly to extract localized features via weighted aggregation like CNNs. Despite the fact that these two types of models start from different places, they fall into the same principle of collecting neighborhood information iteratively to capture high-order correlations among graph nodes and edges. Here "information" is represented as embeddings, *i.e.*, low-dimensional vectors. To this end, the primary and pivotal operation of GNN is to propagate embeddings on graphs following structural connections, including aggregating neighborhood embeddings and fusing them with the target (a node or an edge) embedding, to update graph embeddings layer by layer.

In the following, we will introduce several groundbreaking GNN models to elaborate how neural networks are implemented on graphs. The frequently used notations are explained in Table 1.

- **GCN** [72]. This is a typical spectral model that combines graph convolution and neural networks to achieve the graph task of semi-supervised classification. In detail, GCN approximates the filter in convolution by the first order following [53]. Then the node embeddings are updated as follows,

$$\mathbf{H}^{l+1} = \delta(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^l\mathbf{W}^l), \tag{2}$$

  of which the derivation can refer to [171]. $\mathbf{H}^l \in \mathbf{R}^{|V| \times D}$ is the embedding matrix of graph nodes in the l-th layer of convolution, where D is the embedding dimension. Besides, $\tilde{\mathbf{A}} \in \mathbf{R}^{|V| \times |V|}$ is the adjacency matrix of the graph with self-loop, of which each entry $\tilde{\mathbf{A}}_{ij} = 1$ if the node i connects with j or $i = j$; otherwise $\tilde{\mathbf{A}}_{ij} = 0$, and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$.
- **GraphSAGE** [52]. This is a pioneered spatial GNN model that samples neighbors of the target node, aggregates their embeddings, and merges with the target embedding to update.

$$
\begin{aligned}
\mathbf{h}^l_{\mathcal{N}_i} &= \text{AGGREGATE}_l\left(\{\mathbf{h}^l_j, \forall j \in \mathcal{N}_i\}\right), \\
\mathbf{h}^{l+1}_i &= \delta\left(\mathbf{W}^l\left[\mathbf{h}^l_i \| \mathbf{h}^l_{\mathcal{N}_i}\right]\right),
\end{aligned}
\tag{3}
$$

  where $\mathcal{N}_i$ denotes the sampled neighbors of the target node i. The function AGGREGATE has various options, such as MEAN, LSTM [58] and so on.
- **GAT** [144]. This is a spatial GNN model that addresses several key challenges of spectral models, such as poor ability of generalization from a specific graph structure to another and sophisticated computation of matrix inverse. GAT utilizes attention mechanisms to aggregate neighborhood

features (embeddings) by specifying different weights to different nodes. Specifically, the propagation is formulated as follows,

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

$$
\alpha_{ij} = \frac{\exp \left(\text{LeakyReLU} \left(\mathbf{a}^T \left[\mathbf{W}^l \mathbf{h}_i^l \| \mathbf{W}^l \mathbf{h}_j^l \right] \right) \right)}{\sum\limits_{k \in \mathcal{N}_i} \exp \left(\text{LeakyReLU} \left(\mathbf{a}^T \left[\mathbf{W}^l \mathbf{h}_i^l \| \mathbf{W}^l \mathbf{h}_k^l \right] \right) \right)},
\tag{4}
$$

where α_{ij} is the propagation weight from node j to node i and $\mathcal{N}_i$ is the neighborhood set of node i, including i itself. As shown in the second equation, the attention mechanism is implemented via a fully-connected layer parameterized by a learnable vector $\mathbf{a}$, followed by the *softmax* function.

- **HetGNN** [195]. This is a spatial GNN tailored for heterogeneous graphs. Considering the heterogeneous graph consists of multiple types of nodes and edges, HetGNN first divides neighbors into subsets according to their types. Hereafter, an aggregator function is conducted for each type of neighbor to gather localized information, combining LSTM and MEAN operations. Furthermore, different types of neighborhood information are aggregated based on the attention mechanism. Detailed formulas are omitted since the implementation is following the above works.
- **HGNN** [38]. This is a spectral model implementing GNN on the hypergraph. The convolution is defined as follows,

$$
\mathbf{H}^{l+1} = \mathbf{D}_v^{-\frac{1}{2}} \mathbf{E} \mathbf{D}_e^{-1} \mathbf{E}^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H}^l \mathbf{W}^l,
\tag{5}
$$

where each entry $\mathbf{E}_{ui}$ of $\mathbf{E} \in \mathbf{R}^{|V| \times |E|}$ denotes whether the hyperedge u contains the node i, each diagonal entry of $\mathbf{D}_v \in \mathbf{R}^{|V| \times |V|}$ denotes how many hyperedges the node is included in and each diagonal entry of $\mathbf{D}_e \in \mathbf{R}^{|E| \times |E|}$ denotes how many nodes the hyperedge includes. Generally, this convolution operation can be considered as two stages of propagating neighborhood embeddings: **1)** propagation from nodes to the hyperedge connecting them, and **2)** propagation from hyperedges to the node they meet.

The commonality and difference among typical GNN models above is illustrated in Fig. 11.

In order to further capture high-order structural information on the graph, the convolution or embedding propagation mentioned above will be performed for L times. In most cases, $L \leq 4$ since GNNs are suffered from the over-smoothing problem that the updated embeddings will be in small fluctuations when the number of propagation layers becomes larger. Relevant studies focusing on developing deep and effective GNN models will be introduced in section 5.1.

2.2.3 Model Optimization. After the processing of the designed network in section 2.2.2, overall embeddings of nodes or edges encoding feature semantics as well as graph structures are produced. To perform the downstream graph learning tasks, these embeddings will be further transformed to targets (*e.g.*, the probability that a node belongs to a class) by general neural networks (*e.g.*, MLP).

There are mainly classification, prediction and regression tasks on graphs, including three levels: node, edge, and subgraph. Despite the disparity of various tasks, there is a standard procedure for model optimization. Specifically, relevant embeddings will be mapped and come with labels to formulate the loss function, and then existing optimizers are utilized for model learning. Following this process, there are several types of mapping functions (*e.g.*, MLP, inner product) and loss functions (*e.g.*, pair-wise, point-wise) to choose for specific tasks. For pair-wise loss function, the discrimination between positive and negative samples is encouraged, and a typical formulation

Model	Graph type of GNN	GNN category	Aggregate
GCN		Spectral	
GraphSAGE		Spatial	
GAT		Spatial	
HetGNN		Spatial	
HGNN		Spectral	

Fig. 11. Comparison of several typical GNN models. For graph type, nodes and edges type are represented by colors and line-styles respectively. For aggregation, line-width indicates neighborhood weight.

BPR [126] is

$$\mathcal{L} = \sum_{p,n} -\ln\sigma(s(p) - s(n)), \tag{6}$$

where $\sigma(\cdot)$ is the sigmoid function. p and n denote positive and negative samples respectively, and $s(\cdot)$ is for measuring the samples. For point-wise loss function, it includes mean square error (MSE) loss, cross-entropy loss and so on.

For a better understanding, we take link prediction and node classification tasks as examples to elaborate on how the GNN model is optimized. For link prediction, the likelihood that whether an edge exists between two nodes i, j requires definition. Technically, it is usually calculated based on similarity with node embeddings in each layer of propagation:

$$s(i, j) = f(\{\mathbf{h}_i^l\}, \{\mathbf{h}_j^l\}), \tag{7}$$

where $f(\cdot)$ denotes the mapping function. Furthermore, we can construct the training data as $O = \{(i, j, k)\}$, consisting of observed positive and randomly-selected negative samples, (i, j) and (i, k), respectively. Specifically, the node i connects with j on the graph but not with k. In recommender system, the samples will indicate that user i has interacted with item j but not interacted with item k, where k is sampled from all the other items u has not interacted before. Hereafter, if the pair-wise loss BPR is chosen, the optimization object will be

$$\mathcal{L} = \sum_{(i,j,k) \in O} -\ln\sigma(s(i, j) - s(i, k)). \tag{8}$$

In terms of node classification, node embedding will be transformed to a probability distribution representing which class it belongs to, shown as follows,

$$\mathbf{p}_i = f(\{\mathbf{h}_i^l\}), \tag{9}$$

where $\mathbf{p}_i \in \mathbf{R}^{C \times 1}$ is the distribution and C is the number of classes. Similarly, the training data will be $O = \{(i, \mathbf{y}_i)\}$, where $\mathbf{y}_i \in \mathbf{R}^{C \times 1}$ and i belongs to the class c means $\mathbf{y}_{ic} = 1$; otherwise $\mathbf{y}_{ic} = 0$. In general, the point-wise loss function is usually chosen for the classification task, such as cross-entropy loss which is formulated as

$$\mathcal{L} = - \sum_{(i, \mathbf{y}_i) \in O} \mathbf{y}_i^T \log \mathbf{p}_i. \tag{10}$$

In short, optimization in GNN-based models treats representations generalized by GNNs as input and graph structures (*e.g.*, edges, node classes) as labels, and loss functions are defined for training.

2.3 Why are GNNs required for recommender systems

Over the past decade, recommender systems have evolved rapidly from traditional factorization approaches to advanced deep neural networks based models. Particularly, GNN based recommenders have achieved the state-of-the-art in many aspects, including different recommendaiton stages, scenarios, objectives, applications. The success of GNN based recommenders can be explained from the following three perspectives: (1) structural data; (2) high-order connectivity; (3) supervision signal.

Structural data. Data collected from online platforms comes in many forms, including user-item interaction (rating, click, purchase, etc.), user profile (gender, age, income, etc.), item attribute (brand, category, price, etc.), and so on. Traditional recommender systems are not capable of leveraging those multiple forms of data, and they usually focus on one or a few specific data sources, which leads to sub-optimal performance since much information is ignored. By expressing all the data as nodes and edges on a graph, GNN provides a unified way to utilize available data. Meanwhile, GNN shows strong power in learning representations, and thus, high-quality embeddings for users, items, and other features can be obtained, which is critical to the recommendation performance.

High-order connectivity. Recommendation accuracy relies on capturing similarity between users and items, and such similarity is supposed to be reflected in the learned embedding space. Specifically, the learned embedding for a user is similar to embeddings of items that are interacted by the user. Furthermore, those items that are interacted by other users with similar preferences are also relevant to the user, which is known as the collaborative filtering effect, and it is of great importance for recommendation accuracy. In traditional approaches, the collaborative filtering effect is only implicitly captured since the training data is mainly interaction records that only contain directly connected items. In other words, only first-order connectivity is taken into consideration. The absence of high-order connectivity can largely damage recommendation performance. In contrast, GNN based models can effectively capture high-order connectivity. Specifically, the collaborative filtering effect can be naturally expressed as multi-hop neighbors on the graph, and it is incorporated in the learned representations through embedding propagation and aggregation.

Supervision signal. Supervision signals are usually sparse in the collected data, while GNN based model can leverage semi-supervised signals in the representation learning process to alleviate this problem. Take the E-Commerce platform as an example; the target behavior, purchase, is pretty sparse compared to other behaviors. Therefore, recommender systems that only use the target behavior may achieve poor performance. GNN based models can effectively incorporate multiple non-target behaviors, such as search and add to cart, by encoding semi-supervised signals over the graph, which can significantly improve recommendation performance [67]. Meanwhile, self-supervised signals can also be utilized by designing auxiliary tasks on the graph, which further improves recommendation performance.

3 CHALLENGES OF APPLYING GNNS TO RECOMMENDER SYSTEMS

Although it is well motivated to apply graph neural networks in recommender systems, there exist four parts of critical challenges.

- How to construct appropriate graphs for specific tasks?
- How to design the mechanism of information propagation and aggregation?
- How to optimize the model?
- How to ensure the efficiency of model training and inference?

In the following, we will elaborate on the four challenges one by one.

3.1 Graph Constrcution

Obviously, the first step of applying graph neural networks is to construct graphs. This is in two folds: constructing the data input as graph-structured data; reorganize the recommendation goal as a task on the graph. Taking the task of standard collaborative filtering as an example, the data input is the observed user-item interaction data, and the output is predictions of the missing user-item interactions. Therefore, a bipartite graph with users/items as nodes and interactions as edges can be constructed. Besides, the CF task turns to the user-item link prediction on the graph.

However, it is challenging to construct graphs that can well handle the task properly. It should be carefully implemented with the consideration of the following aspects.

- **Nodes.** One of the main goals of learning with graph neural networks is to assign nodes the representations. This results in that the definition of nodes largely determines the scale of the GNN models, of which the majority of parameters are occupied by the layer-0 node embeddings. Note that edge embeddings are usually either not considered or computed based on node embeddings. On the other hand, it is also a challenging problem to determine whether to distinguish different types of nodes. For example, in the collaborative filtering task, user nodes and item nodes can be modeling differently or considered as the same kind of nodes. Another challenging point is to handle concrete input such as some numerical features like item prices, which are always continuous numbers. To represent these features in the graph, one possible solution is to discretize them to categorical ones, which can be then represented as nodes [207].
- **Edges.** The definition of edges highly affects the graph's quality in further propagation and aggregation, along with the model optimization. In some trivial tasks, the data input of the recommender system can be considered as a kind of relational data such as user-item interactions or user-user social relations. In some complex tasks, other relations can also be represented as edges. For example, in bundle recommendation, a bundle consists of several items. Then the edge connecting the bundle and item can reflect the relation of affiliation. Good designs of edges when constructing a graph should fully consider the graph density. A too-dense graph means there are nodes with extremely high degrees. This will make the embedding propagation conducted by a very large number of neighbors. It will further make the propagated embedding non-distinguished and useless. To handle too dense edges, sampling, filtering, or pruning on graphs are promising solutions. A too-sparse graph will also, of course, results in the poor utility of embedding propagation since the propagation will be conducted on only a small fraction of nodes.

3.2 Network Design

What makes GNN different from the traditional graph learning methods is the propagation layer. As for the propagation, how to choose the path is critical for modeling the high-order similarity in recommender systems. Besides, the propagation can also be parametric, which assigns different weights to different nodes. For example, when propagating item embeddings to a user node in the

user-item interaction graph, it captures the item-based CF effect. The weights refer to the different importance of historically interacted items.

In the propagation, there are also various choices of aggregation functions, including mean pooling, LSTM, max, min, etc. Since there is no single choice that can perform the best among all recommendation tasks or different datasets, it is vital to design a specific and proper one. Besides, the different choices of propagation/aggregation highly affect the computation efficiency. For example, mean pooling is widely used in GNN-based recommendation models since it can be computed efficiently, especially for the graph containing high-degree nodes, such as very popular items (which can connect a large number of users). Also, the propagation/aggregation layers can be stacked to help nodes access higher-hops neighbors. Too shallow layers make the high-order graph structure cannot be well modeled, and too deep ones make the node embedding over-smoothed. Either one of the two cases will lead to poor recommendation performance.

3.3 Model Optimization

To optimize the graph neural network-based recommendation models, the traditional loss functions in recommender system always turn to graph learning losses. For example, the *logloss* in the optimization can be regarded as the point-wise link prediction loss. Similarly, BPR loss [126] is usually adopted in the link prediction task on graphs. Another aspect is data sampling. In GNN-based recommendation, to sample positive or negative items, the sampling manner can highly depend on the graph structure. For example, in social recommendation, performing random walk on the graph can generate weak positive items (such as items interacted by friends).

In addition, sometimes, GNN-based recommendation may involve multiple tasks, such as the link prediction tasks on different types of edges. Then in such a case, how to balance each task and make them enhance each other is challenging.

3.4 Computation Efficiency

In the real world, recommender systems should be trained/inferred efficiently. Therefore, to ensure the application value of GNN-based recommendation models, its computation efficiency should be seriously considered. Compared with traditional non-GNN recommendation methods such as NCF or FM, GNN models' computation cost is far higher. Especially for the spectral GNN models such as GCN, complex matrix operations are involved in each GCN layer. With multi-layer stacking of GCN layers, the computation cost further increases. Therefore, spatial GNN models such as PinSage can be easier to be implemented in large-scale industrial applications. With sampling among neighbors or pruned graph structure, efficiency can always be kept as long as we can bear the drop of recommendation performance.

4   EXISTING METHODS

4.1   Taxonomy

In recent years, GNN has been applied to a wide range of recommendation tasks. Here we define the taxonomy in terms of recommendation stages, scenarios, objectives, and applications, respectively. To be more specific, recommendation stages indicate the overall procedure that a recommender system is implemented in the real-world platform. The procedure includes matching for item candidates selection, ranking for capturing user preferences, and re-ranking for other criteria beyond accuracy. The recommendation scenarios including social recommendation, sequential recommendation, cross-domain recommendation, etc. The recommendation objectives incorporate accuracy, diversity, explainability, fairness, and so on, in which accuracy is of the most concern. The recommendation application refers to specific industrial applications. Table 2, 3, and 4 show

Table 2. A summary of GNN-based models in different recommendation stages in top-tier venues.

Perspective	Category	Model	Venue	Year
Stage	Matching	GCMC [8]	KDD	2018
		PinSage [188]	KDD	2018
		NGCF [153]	SIGIR	2019
		LightGCN [55]	SIGIR	2020
	Ranking	Fi-GNN [87]	CIKM	2019
		PUP [207]	ICDE	2020
		L_0-SIGN [135]	AAAI	2021
		DG-ENN [50]	KDD	2021
	Re-ranking	IRGPR [94]	CIKM	2020

representative researches of GNN-based recommendation published in top-tier venues for different recommendation stages, recommendation scenarios, and recommendation objectives, respectively.

4.2 GNN in Different Recommendation Stages

4.2.1 GNN in Matching. In the matching stage, efficiency is an essential problem because of the high computation complexity for candidates selection. Specifically, only hundreds of items will be selected from the item pool of million magnitudes for the following ranking stage, based on coarse-grained user preferences. Therefore, proposed models in this stage barely leverage user-item interactions as data input for modeling user preferences, without introducing additional features such as user ages, item price, browsing time on the application, etc.

GNN-based models in the matching stage can be regarded as embedding matching, usually designing specialized GNN architecture on the user-item bipartite graph [8, 137, 153, 154, 161]. Berg *et al.* [8] proposed to pass neighborhood messages by summing and assign weight-sharing transformation channels for different relational edges (*i.e.*, user-item ratings). Wang *et al.* [153] proposed a spatial GNN in recommendation and obtain superior performance compared with conventional CF methods like MF [74] or NCF [56]. Sun *et al.* [137] argued that simple aggregation mechanisms like sum, mean, or max, cannot model relational information among neighbors and proposed a neighbor interaction-aware convolution to address the issue. Wang *et al.* [154] developed disentangled GNN to capture independent user intentions, which extends the set of candidate items in matching and guarantees the accuracy simultaneously. Wu *et al.* [161] leverage the stability of graph structure to incorporate a contrastive learning framework to assist representation learning. These GNN-based models are capable of capturing high-order similarity among users and items as well as structural connectivity. In this way, the semantics that users with similar interactions will have similar preferences are extended through multiple times of information propagation. On the other hand, the training complexity of GNN-based models were demonstrated [153, 154] acceptable and comparable with non-graph models, especially when the transformation matrix is removed [55]. Besides, [188] showed that GNN-based model could be applied to web-scale recommender system in real-world platforms efficiently and effectively, which combines random walk and GraphSAGE [52] for embedding learning on a large-scale item-item graph. Table 5 shows the commonality and difference among the GNN models in matching.

In a nutshell, GNN can be applied to recommendation tasks effectively, which balances the accuracy and efficiency of generating candidates from the item pool.

4.2.2 GNN in Ranking. In the ranking stage, with a much smaller amount of candidate items, more accurate models can be utilized, and more features can be included. Existing ranking models usually first transform sparse features into one-hot encodings then transform them into dense

Table 3. A summary of GNN-based models in different recommendation scenarios in top-tier venues.

Perspective	Category	Model	Venue	Year
Scenario	Social	DiffNet [164]	SIGIR	2019
		GraphRec [35]	WWW	2019
		DANSER [167]	WWW	2019
		DGRec [134]	WSDM	2019
		HGP [71]	RecSys	2019
		DiffNet++ [163]	TKDE	2020
		MHCN [193]	WWW	2021
		SEPT [191]	KDD	2021
		GBGCN [196]	ICDE	2021
		KCGN [63]	AAAI	2021
		DiffNetLG [132]	SIGIR	2021
	Sequential	ISSR [91]	AAAI	2020
		MA-GNN [100]	AAAI	2020
		STP-UDGAT[90]	CIKM	2020
		GPR[12]	CIKM	2020
		GES-SASRec[215]	TKDE	2021
		RetaGNN[59]	WWW	2021
		TGSRec[36]	CIKM	2021
		SGRec[85]	IJCAI	2021
		SURGE [14]	SIGIR	2021
	Session	SR-GNN [169]	AAAI	2019
		GC-SAN [181]	IJCAI	2019
		TA-GNN [190]	SIGIR	2020
		MGNN-SPred [150]	WWW	2020
		LESSR [23]	KDD	2020
		MKM-SR [109]	SIGIR	2020
		GAG [123]	SIGIR	2020
		GCE-GNN [157]	SIGIR	2020
		SGNN-HN [116]	CIKM	2020
		DHCN [176]	AAAI	2021
		SHARE [148]	SDM	2021
		SERec [24]	WSDM	2021
		COTREC [175]	CIKM	2021
		DAT-MID [15]	SIGIR	2021
		TASRec [211]	SIGIR	2021
	Bundle	BGCN[13]	SIGIR	2020
		HFGN[83]	SIGIR	2020
		BundleNet[33]	CIKM	2020
		DPR[209]	WWW	2021
	Cross Domain	PPGN[204]	CIKM	2019
		BiTGCF[92]	CIKM	2020
		DAN[146]	CIKM	2020
		HeroGRAPH[30]	Recsys	2020
		DAGCN[49]	IJCAI	2021

embedding vectors. These feature embeddings are directly concatenated and fed into DNN [26, 48] or specifically designed models [56, 125, 133] in an unstructured way to estimate the ranking score. The main challenge of utilizing GNN for ranking is designing proper structures to capture feature interactions. Specifically, GNN-based ranking models usually consist of two components, encoder and predictor, which address feature interaction from different directions. On the one hand, special graph structures can be designed in the encoder to capture the desired feature interactions. On the

Table 4. A summary of GNN-based models for different recommendation objectives in top-tier venues.

Perspective	Category	Model	Venue	Year
Objective	Multi-behavior	MBGCN [67]	SIGIR	2020
		MGNN-SPred[150]	WWW	2020
		MGNN[200]	CIKM	2020
		LP-MRGNN[151]	TKDE	2021
		GNMR[172]	ICDE	2021
		MB-GMN[174]	SIGIR	2021
		KHGT[173]	AAAI	2021
		GHCF[16]	AAAI	2021
		DMBGN[178]	KDD	2021
	Diversity	V2HT [82]	CIKM	2019
		BGCF [136]	KDD	2020
		DGCN [205]	WWW	2021
	Explainability	RippleNet [147]	CIKM	2018
		EIUM [65]	MM	2019
		KPRN [155]	AAAI	2019
		RuleRec [101]	WWW	2019
		PGPR [177]	SIGIR	2019
		KGAT [152]	KDD	2019
		TMER [20]	WSDM	2021
	Fairness	FairGo [162]	WWW	2021
		FairGNN [31]	WSDM	2021

other hand, feature interaction can be taken into consideration in the predictor, where the ranking score is estimated by integrating different feature embeddings from the GNN encoder.

Li *et al.* [87] propose Feature Interaction Graph Neural Networks (Fi-GNN), which constructs a weighted fully connected graph of all the input features. The encoder in Fi-GNN is composed of a GAT and a GRU, and the predictor is achieved with attention networks. Zheng *et al.* [206, 207] investigate the influence of price feature in ranking and propose a model called Price-aware User Preference modeling (PUP). They design an encoder with GCN on a pre-defined heterogeneous graph to capture price awareness, and a two-branch factorization machine is utilized as the predictor. Since not all feature interactions are useful, L_0-SIGN [135] automatically detects beneficial feature interactions and only reserves those edges, resulting in a learned graph which is further fed into a graph classification model to estimate the ranking score. In addition, Guo *et al.* [50] propose DG-ENN with a dual graph of an attribute graph and a collaborative graph, which integrates the information of different fields to refine the embedding for ranking. Furthermore, SHCF [79] and GCM [160] utilize extra nodes and edge attributes to represent item attributes and context information, respectively. Classical interaction predictors are adopted, such as inner product and FM. Table 6 illustrates the differences of the above GNN based ranking models in terms of the designs of encoder and predictor.

4.2.3 GNN in Re-ranking. After obtaining the scores of recommended items, top items are further re-ranked with pre-defined rules or functions to improve the recommendation quality. Specifically, two key factors need to be considered in re-ranking. First, different items can have mutual influence by certain relationships such as substitutability and complementarity. Second, different users tend to have distinct preferences, and thus re-ranking can also be personalized. GNN provides a unified way to encode both item relationships and user preferences. Liu *et al.* [94] propose a model called IRGPR to accomplish personalized re-ranking with the help of GNN. They propose a heterogeneous graph to fuse the two information sources, one item relation graph to capture multiple item relationships,

Table 5. Details of GNN models in matching stage.

Model	Graph	GNN	Motivations
GCMC [8]	user-item	GCN	weight sharing among relations
NGCF [153]	user-item	GCN	-
DGCF [154]	user-item	GAT	disentangled representations
LightGCN [55]	user-item	LightGCN	remove transformation and nonlinearity
SGL [161]	user-item	GCN	self-supervision on graphs
NIA-GCN [137]	user-item	NIA-GCN	neighbor interaction (NI)
PinSage [188]	item-item	GCN	sampling

Table 6. Details of GNN models in ranking stage.

Model	Graph	GNN	Predictor
Fi-GNN [87]	fully-connected	GAT+GRU	attention
PUP [207]	pre-defined	GCN	two-branch FM
L_0-SIGN [135]	learned	SIGN	graph classification
DG-ENN [50]	pre-defined	LightGCN	DNN
SHCF [79]	pre-defined	HGAT	inner product
GCM [160]	pre-defined	context GCN	FM

and one user-item scoring graph to include the initial ranking scores. User and item embeddings are obtained after multiple message propagation layers, including global item relation propagation and personalized intent propagation. The final order of re-ranked items is generated with a feed-forward network.

4.3 GNN in Different Recommendation Scenarios

4.3.1 GNN in Social Recommendation. In social recommendation, we have social networks that contain the social relations of each user, and the goal is to utilize the local neighbors' preferences for each user in social networks to enhance the user modeling [17, 138, 165, 166]. From the perspective of representation learning with GNN, there are two key considerations in social recommendation: 1) how to capture the social factor; 2) how to combine the social factor from friends and user preference from his/her interaction behaviors. Here, we first summarize how the existing works capture the social factors from two perspectives, *i.e.*, graph construction, and information propagation.

- **Graph construction.** In social-aware recommender systems, user's final behavior is decided by both the social impacts from friends and his/her own preferences. One of the main challenges in social recommendation is how to construct a social graph to capture the social influences from friends. Generally speaking, a certain user in social networks is not only influenced by his/her friends (the first-order neighbors) but also influenced by friends' friends (the high-order neighbors). To capture the high-order social relations, the graph construction methods can be divided into two directions: stacked graphs and hypergraph. Given that normal graph can only model the pairwise relations, normal graph-based methods [35, 51, 71, 98, 113, 132, 163, 164, 167, 184, 191, 192] stacked multiple GNN layers to capture multi-hop high-order social relations. However, stacked GNN layers may suffer from the over-smoothing [18] problem, which may lead to significant performance degradation. Hypergraph-based methods, such as MHCN [193], propose to model the high-order social relations with hyperedge [38], which can connect more than two nodes and model the high-order relations in a natural way. HOSR [96]
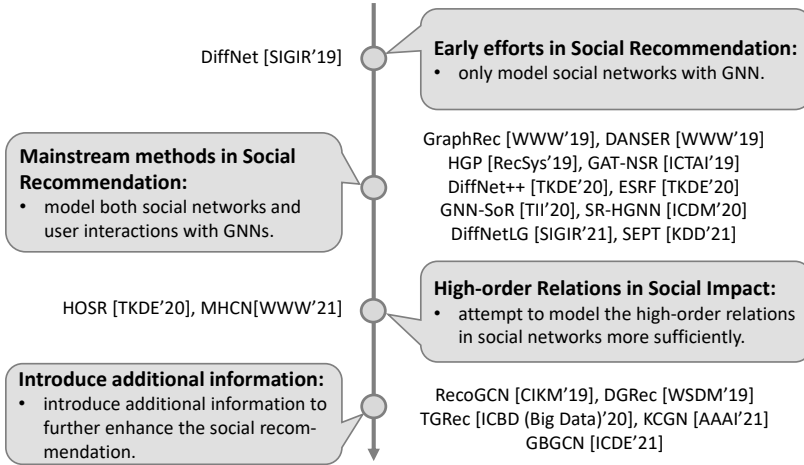
Fig. 12. Illustration of GNN models for social recommendation.

recursively propagate embeddings along the social network for reflecting the influence of high-order neighbors in the user representations. To further improve the recommendation performance, some works [6, 63, 134, 182, 196] introduce side information when constructing the graph. RecoGCN [182] unifies users, items, and selling agents into a heterogeneous graph to capture the complex relations in social E-commerce. GBGCN [196] constructs a graph for organizing user behaviors of two views in group-buying recommendation, of which the *initiator view* contains those initiator-item interactions and the *participant view* contains those participant-item interactions. DGRec [134] and TGRec [6] introduce temporal information of user behaviors into social recommendation. KCGN [63] proposes to capture both user-user and item-item relations with the developed knowledge-aware coupled graph.

- **Information propagation.** As for the propagation on the constructed graph for social recommendation, there are two main propagation mechanisms, *i.e.*, average-pooling (GCN) and attention mechanism (GAT). The methods with average-pooling mechanism [6, 51, 63, 71, 132, 164, 184, 191, 193, 196] conduct average-pooling propagation (GCN) on social graph and treats the social influence of friends equally. RecoGCN [182] conducts meta-path based GCN propagation on the constructed graph to capture both the social impact and user preference. HOSR [96] aggregates the information from neighbors with GCN to capture the high-order relations in social graph. MHCN [193] performs propagation with GCN on constructed hypergraph to obtain the high-order social relations. The methods with attention mechanism [35, 98, 113, 134, 163, 167, 192], such as GraphRec [35] and DiffNet++ [163], assume that the social influences from different neighbors on the social graph are different and assign different weights to the social influences from different friends.

In social recommendation, user representations are learned from two distinct perspectives, *i.e.* social influence and user interactions. To combine the user representations from the above two perspectives, there are two strategies, 1) separately learn user representations from the social graph and user-item bipartite graph and 2) jointly learn user representations from the unified graph that consists of social graph and user-item bipartite graph. The methods with the first strategy, such as DiffNet [164], GraphRec [35] and MHCN [193], first separately learn user representations from social graph and user-item graph, and then combines the representations with sum-pooling [164, 193],

Table 7. Details of GNN models for social recommendation.

Model	Graph	GNN	Social Signal Extraction
DiffNet [164]	social graph	GCN	sum-pooling
GraphRec [35]	social graph + user-item graph	GAT	concatenation
DANSER [167]	social graph + user-item graph	GAT & GCN	-
DiffNet++ [163]	heterogeneous graph	GAT	multi-level attention network
MHCN [193]	multi-channel hypergraph + user-item graph	HyperGCN	sum-pooling
SEPT [191]	triangle-graphs + user-item graph	GCN	-
RecoGCN [182]	heterogeneous graph	Meta-path + GCN	concatenation
ESRF [192]	motif-induced graph	GAT	sum-pooling
GNN-SoR [51]	heterogeneous graph	GCN	concatenation
ASR [98]	heterogeneous graph	GAT	concatenation
GBGCN [196]	heterogeneous graph	GCN	-
DGRec [134]	social graph	GAT	-
SR-HGNN [184]	social graph + user-item graph	GCN	concatenation
KCGN [63]	social graph + item-item graph	GCN	concatenation
HGP [71]	group-user graph + user-item graph	GCN	attention mechanism
GAT-NSR [113]	social graph + user-item graph	GAT	MLP
HOSR [96]	social graph + user-item graph	GCN	attention mechanism
DiffNetLG [132]	heterogeneous graph	GCN	concatenation
TGRec [6]	heterogeneous graph	GCN	attention mechanism

concatenation [35], MLP [113] or attention mechanism [6, 71, 96]. DiffNet++ [163], a typical method with the second strategy, first aggregates the information in the user-item sub-graph and social sub-graph with the GAT mechanism and then combines the representations with the designed multi-level attention network at each layer. Table 7 shows the differences among the above approaches for social recommendation.

To sum up, the development of social recommendation with GNN can be summarized in Fig. 12. Early efforts in Social Recommendation only model the social network with GNN, such as DiffNet [164]. Then, the methods [35, 51, 71, 98, 113, 132, 163, 164, 167, 184, 191, 192] that model both social network and user interactions with GNNs become the mainstream in GNN-based social recommendation. Moreover, some studies, such as MHCN [193] and HOSR [96], attempt to enhance the recommendation by modeling the high-order relations in social networks more sufficiently. Also, there exist some works [6, 63, 134, 182, 196] that introduce additional information to further enhance the social recommendation.

4.3.2 GNN in Sequential Recommendation. For sequential recommendation, in order to improve the recommendation performance, it is necessary to extract as much effective information as possible from the sequence, and to learn the user's interest in the sequence, including short-term interest, long-term interest, dynamic interest, etc., so as to accurately predict the next item that the user may be interested in. Some tools for sequence modeling have been used, such as Markov chains [25] or recurrent neural networks [69]. For graph neural networks, it can be well leveraged for short-term, dynamic interest modeling or representation learning in by converting the data to graph. A general pattern for sequential modeling with GNN is shown in figure 13.

SURGE [14] transforms the sequence of each user into an item-item graph and adaptively learns the weights of edges through metric learning, with only the stronger edges retained by dynamic graph pooling. The retained graph is converted to a sequence by position flatten and finally be used to predict the next item. Ma *et al.* [100] considers the short-term interest modeling in the sequence to build an item-item graph. For each item, it only builds edges with other items close to it in the sequence. This enables it to learn short-term user interests in the sequence while still
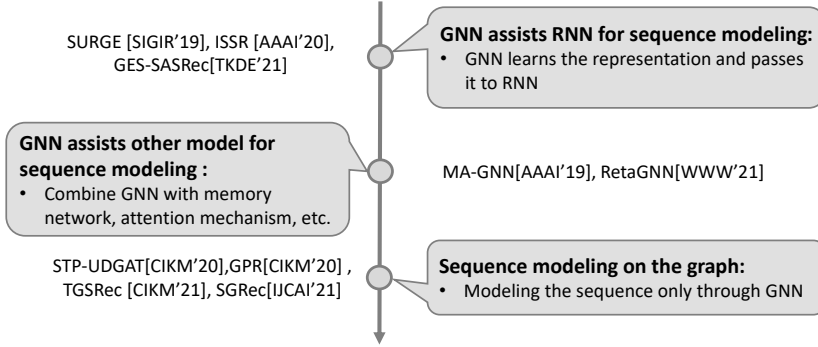
SURGE [SIGIR'19], ISSR [AAAI'20],
GES-SASRec[TKDE'21]

GNN assists RNN for sequence modeling:
• GNN learns the representation and passes it to RNN

GNN assists other model for sequence modeling :
• Combine GNN with memory network, attention mechanism, etc.

MA-GNN[AAAI'19], RetaGNN[WWW'21]

STP-UDGAT[CIKM'20],GPR[CIKM'20] ,
TGSRec [CIKM'21], SGRec[IJCAI'21]

Sequence modeling on the graph:
• Modeling the sequence only through GNN

Fig. 13. Illustration of GNN models for sequential recommendation.

Table 8. Details of GNN models for sequential recommendation.

Model	Graph	GNN	Sequential Modeling
SURGE [14]	item-item graph	GAT	RNN
[145]	item-item graph	GCN	Attention
ISSR[91]	item-item and user-item graph	GCN	RNN
MA-GNN[100]	item-item graph	GCN	Memory network
DGSR[199]	user-item graph	GAT	RNN
GES-SASRec[215]	item-item graph	GCN	RNN
RetaGNN[59]	temporal heterogeneous graph	GAT	Self Attention
TGSRec[36]	temporal user-item graph	GAT	GAT
SGRec[85]	item-item graph	GAT	GAT
GME[179]	item-item and user-item graph	GAT	GAT
STP-UDGAT[90]	item-item and user-item graph	GAT	GAT
GPR[12]	item-item and user-item graph	GCN	GCN

learning long-term user interests through other networks. The learned multiple representations are fused together and used for final recommendation.

Since GNN has the ability of high-order relationship modeling by aggregating information from neighbor nodes, after fusing multiple sequences into one graph, it can learn representations of both users and items in different sequences, which can't be accomplished by Markov model or recurrent neural network. Wang *et al.* [145] propose a simple method that directly converts the sequence information into directed edges on the graph and then uses GNN to learn representations. Liu *et al.* [91] construct a user-item bipartite graph and an item-item graph at the same time, where the edges of the item-item graph indicate co-occurrence in a sequence, with edge weights assigned according to the number of occurrences. The representations learned by GNN are used in the final recommendation through the recurrent neural network. Different from directly converting the temporal sequence into directed edges in the graph, DGSR [199] and TGSRec [36] consider the timestamps in the sequence in the process of graph construction. In the graph, each edge represents the interaction between the user and the item, and has the corresponding time attribute. Then perform convolution operations on the temporal graph to learn the representations of users and items. GES-SASRec [215] and SGRec [85] focus on the learning of item representations. For an item in a sequence, GES-SASRec [215] considers the next item of this item in other sequences, and SGRec [85] not only considers the next item but also considers the previous one. By aggregating
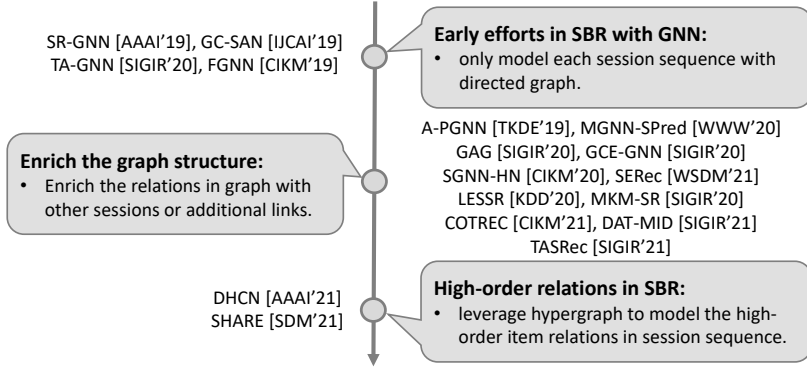
Fig. 14. Illustration of GNN models for session-based recommendation (SBR).

Table 9. Details of GNN models for session-based recommendation.

Model	Graph	GNN	Enrich Graph Structure
SR-GNN [169]	directed graph	gated GNN	-
GC-SAN [181]	directed graph	gated GNN	-
TA-GNN [190]	directed graph	gated GNN	-
FGNN [122]	directed graph	GAT	-
A-PGNN [170]	directed graph	gated GNN	cross sessions
MGNN-SPred [150]	directed & multi-relational item graph	GraphSAGE	cross sessions
GAG [123]	directed graph	GCN	cross sessions
SGNN-HN [116]	star graph	gated GNN	additional edges
GCE-GNN [157]	directed graph + global graph	GAT	cross sessions
DGTN [208]	directed graph	GCN	cross sessions
DHCN [176]	hypergraph + line graph	HyperGCN	cross sessions
SHARE [148]	session hypergraph	HyperGAT	additional edges
SERec [24]	KG + directed graph	GAT + gated GNN	cross sessions
LESSR [23]	directed graph	GAT	additional edges
CAGE [128]	KG + article-level graph	GCN	cross sessions
MKM-SR [109]	KG + directed graph	gated GNN	cross sessions
COTREC [175]	item graph + line graph	GCN	cross sessions
DAT-MDI [15]	directed graph + global graph	GAT	cross sessions
TASRec [211]	dynamic graph	GCN	cross sessions

the items before and after the target item in different sequences, the representation of the item is enhanced. GPR [12] and GME [179] constructs edges between items by considering the frequency of consecutive occurrences or occurrences in the same sequence to enhance the representation. Some works are more complicated. For example, RetaGNN [59] considers the attributes of the items when constructing the graph, while STP-UDGAT [90] considers the geographic location, timestamp, and frequency in the POI recommendation. Table 8 summarizes the above works.

4.3.3 GNN in Session-based Recommendation . In session-based recommendation, the session data may contain both user interersts and noisy signals. Suppose a session for a certain user, *iPhone* → *iPad* → *milk* → *AirPods*. Obviously, *milk* is likely clicked by mistake and then becomes a noise, as the session reflects the user's preference for electronic products. Hence, the two main considerations in session-based recommendation are 1) how to model the item transition pattern in session data,

and 2) how to activate user's core interests from noisy data. From the perspective of graph learning, the item transitions can be modeled as graph, and the information propagation on the graph can activate user's actual interests.

- **Graph construction.** In session-based recommendation, most existing works [122, 157, 169, 181, 190] model the session data with a directed graph to capture the item transition pattern. Distinct from sequential recommendation, the session sequence in session-based recommendation is short and the user behaviors are limited, *i.e.*, the average length of sequences in Tmall‖ is only 6.69 [157, 176]. Hence, a session graph constructed from a single session may only contain limited nodes and edges. To address the above challenge and sufficiently capture the possible relations among items, there are two strategies, 1) straightforwardly capturing relations from other sessions, and 2) add the additional edges of the session graph. For the first strategy, A-PGNN [170], DGTN [208], and GAG [123] propose to enhance relations of the current session graph with related sessions, and GCE-GNN [157] leverages the global context by constructing another global graph to assist the transition patterns in the current session. DHCN [176] regards each session as a hyperedge and represents all sessions in a hypergraph to model the high-order item relations. SERec [24] enhances the global information for each session with a knowledge graph. CAGE [128] learns the representations of semantic-level entities by leveraging the open knowledge graph to improve the session-based news recommendation. MKM-SR [109] enhances the information in the given session by incorporating user micro-behaviors and item knowledge graph. COTREC [175] unifies all sessions into a global item graph from item view and captures the relations among sessions by line graph from session view. DAT-MID [15] follows GCE-GNN [157] to construct both session graph and global graph and then learns item embeddings from different domains. TASRec [211] constructs a graph for each day to model the relations among items and enhance the information in each session. As for the second strategy, SGNN-HN [116] constructs star graph with a "star" node to gain extra knowledge in session data. SHARE [148] expands the hyperedge connections by sliding the contextual window on the session sequence. LESSR [23] proposes to first construct an edge-order preserving multigraph and then construct a shortcut graph for each session for enriching edge links.

- **Information propagation.** As for the information propagation on the constructed graph, there are four propagation mechanisms that are used in session-based recommendation, e.g. gated GNN, GCN, GAT, and GraphSAGE. SR-GNN [169] and its related works [24, 109, 116, 170, 181, 190] combine the gated recurrent units in the propagation (gated GNN) on the session graph. GAG [123], DCTN [208] conduct graph convolution on the constructed directed graph. DHCN [176] proposes to perform graph convolution on both hypergraph and line graph to obtain session representations from two different perspectives. Similar to DHCN [176], COTREC [175] performs GCN on item graph and line graph to obtain information from item and session views, respectively. CAGE [128] conducts GCN on article-level graph and TASRec [211] performs graph convolution on the dynamic graph to capture the item relations. FGNN [122] conducts GAT on a directed session graph to assign different weights on different items. SHARE [148] performs GAT on session hypergraphs to capture the high-order contextual relations among items. GCE-GNN [157] and DAT-MID [15] perform GAT on both session graph and global graph to capture the local and global information, respectively. MGNN-SPred [150] adopts GraphSAGE on multi-relational item graph to capture the information from different types of neighbors.

Table 9 shows the differences among the above approaches for session-based recommendation. To sum up, the development of session-based recommendation (SBR) with GNN can be summarized in Fig. 14. Early efforts in SBR only model each session sequence with a directed graph, such as
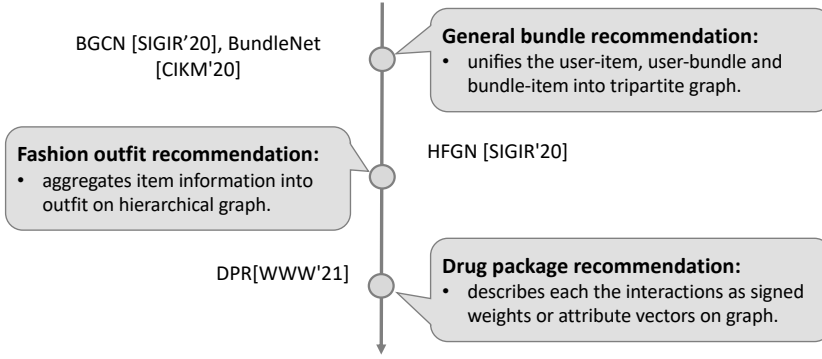

‖https://www.tmall.com

Fig. 15. Illustration of GNN models for bundle recommendation.

Table 10. Details of GNN models for bundle recommendation (how they address challenges).

Model	C1:modeling affiliation	C2: refining bundle	C3: high-order relation
BGCN[13]	tripartite graph	meta-path propagation	multi-layer GCN
HFGN[83]	hierarchical Graph	direct aggregation	multi-layer GCN
BundleNet[33]	tripartite graph	transformation parameters	multi-layer GCN
DPR[209]	tripartite graph	graph induction	multi-layer GCN

SR-GNN [169], GC-SAN [181], TA-GNN [190], and FGNN [122]. Then, some methods attempt to enrich the relation and information in the session graph with other sessions or additional links. The methods in [15, 24, 109, 123, 128, 150, 157, 170, 175, 176, 208, 211] combine the information from other sessions to capture more information from similar sessions or all sessions. Moreover, the methods with additional links, such as SGNN-HN [116], SHARE [148] and LESSR [23], attempt to introduce additional edges to the given session to capture the complex relations and information in session data. Furthermore, some studies, such as DHCN [176] and SHARE [148], attempt to enhance the recommendation by modeling the high-order relations in session data more sufficiently.

4.3.4   GNN in Bundle Recommendation. The three challenges of bundle recommendation are 1) users' decisions towards bundles are determined by the items the bundles contain (the affiliation relation), 2) learning bundle representations with the sparse user-bundle interactions, and 3) high-order relations. The earlier works approach the problem of bundle recommendation by learning from user-item interaction and user-bundle interaction together, with parameter sharing or joint loss function [22, 118]. For the first time, Chang *et al.* [13] propose a GNN-based model that unifies both two parts of interactions and the bundle-item affiliation-relations into one graph. Then the item can serve as the bridge for embedding propagation between user-bundle and bundle-bundle. Besides, a specially designed sampling manner for finding hard-negative samples is further proposed for training. Deng *et al.* [33] construct a similar tripartite graph with the transformation parameters to well extract bundle representations from included items' representations. Zheng *et al.* [209] consider the bundle recommendation in the drug package dataset and propose to initialize a graph with auxiliary data and represent the interaction as scalar weights and vectors. GNN layers are proposed for obtaining the drug package embeddings. Li *et al.* [83] consider the problem of personalized outfit recommendation, which can also be regarded as a kind of bundle recommendation. The authors construct a hierarchical graph, of which the users, items, and outfits are contained. GNN
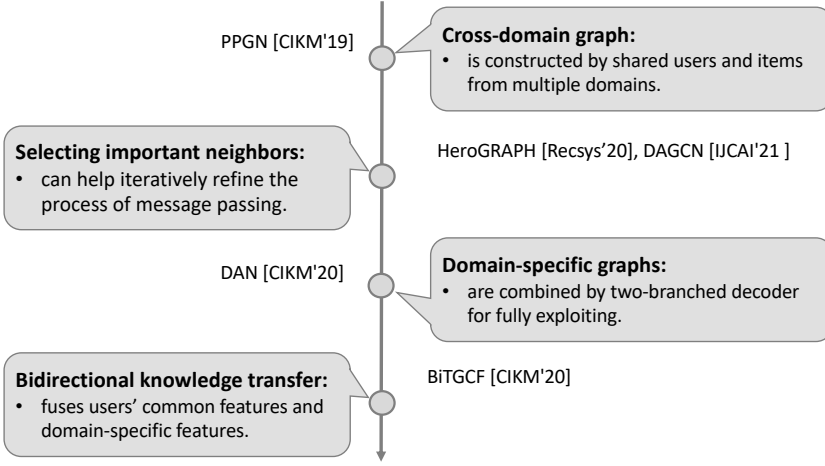
Fig. 16. Illustration of GNN models for cross-domain recommendation.

Table 11. Details of GNN models of cross-domain recommendation.

Model	Graph	Information Transferring
PPGN[204]	cross-domain graph	cross-domain propagation
BiTGCF[92]	domain-specific graphs	common user attributes
DAN[146]	domain-specific graphs	two-branched decoder
HeroGRAPH[30]	cross-domain graph	cross-domain propagation
DAGCN[49]	cross-domain graph	cross-domain propagation

layers are deployed for obtaining the representations of users and outfits. The learning of the GNN model follows a multi-task manner.

In summary, the data input of bundle recommendation can be well represented as graph-structural data, especially for the bundles, which have been represented as a kind of new nodes.

4.3.5 GNN in Cross-Domain Recommendation. Benefit from the powerful capabilities, GNN-based recommendation model has gradually emerged in the field of cross-domain recommendation. Zhao *et al.* [204] construct the cross-domain graph with shared users and items from multiple domains. The proposed PPGN's embedding propagation layers can well learn the users' preferences on multiple domains' items under a multi-task learning framework. Liu *et al.* [92] propose the bidirectional knowledge transfer by regarding the shared users as the bridge. GNN layers are adopted to leverage the high-order connectivity in the user-item interaction graph for better preference learning with the user features, fuses from common features, and domain-specific features. Guo *et al.* [49] construct the graph for each domain and deploy domain-specific GCN layers for learning user-specific embeddings. The authors combine it with the attention mechanisms for adoptively choose important neighbors during the embedding propagation. Wang *et al.* [146] propose an encoder-decoder framework where the encoder is implemented by graph convolutional networks. Specifically, the GCNs are deployed on the user-item interaction graphs. Cui *et al.* [30] proposes to construct a heterogeneous graph, where multiple domains' users and items can be well included. The GNN-based embedding propagation is deployed on multiple domains, where a user/item can directly absorb the information of different domains, where recurrent attention networks are used for distinguishing important neighbors.
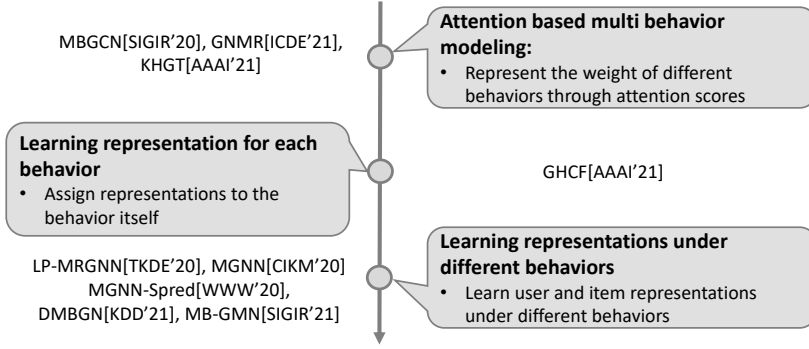
MBGCN[SIGIR'20], GNMR[ICDE'21],
KHGT[AAAI'21]

Attention based multi behavior modeling:
• Represent the weight of different behaviors through attention scores

Learning representation for each behavior
• Assign representations to the behavior itself

GHCF[AAAI'21]

LP-MRGNN[TKDE'20], MGNN[CIKM'20]
MGNN-Spred[WWW'20],
DMBGN[KDD'21], MB-GMN[SIGIR'21]

Learning representations under different behaviors
• Learn user and item representations under different behaviors

Fig. 17. Illustration of GNN models for multi-behavior recommendation.

Table 12. Details of GNN models for multi-behavior recommendation.

Model	Graph	GNN	Multi-Behavior Modeling
MB-GMN[174]	user-item graph	GCN / GAT	Behavior representation
KHGT[173]	user-item / item-item graph	GAT	Weight
MBGCN[67]	user-item graph	GAT	Weight
MGNN-SPred[150]	item-item graph	GCN	Weight
MGNN[200]	user-item graph	GCN	Node representation
LP-MRGNN[151]	item-item graph	GCN	Weight
GNNH[189]	item-item / category-category graph	GCN	Item representation
GNMR[172]	user-item graph	GAT	Weight
DMBGN[178]	item-item graph	GCN	Graph representation
GHCF[16]	user-item graph	GCN	Edge embedding

4.3.6 GNN in Multi-behavior Recommendation. Multiple types of behaviors can provide a large amount of information to the recommender system, which helps the recommender system to learn the user's intentions better, thereby improving the recommendation performance. For recommender systems based on graph neural networks, on the basis of common user-item bipartite graphs, the multiple types of behaviors between users and items can naturally be modeled as different types of edges between nodes. Therefore, most of the multi-behavior recommendation methods based on graph neural networks are based on heterogeneous graphs. However, the focus of multi-behavior recommendation is 1) how to model the relationship between multiple behaviors and target behavior, and 2) how to model the semantics of the item through behavior, which is shown in Fig. 17.

In order to model the effect of auxiliary behaviors on target behaviors, the simplest method is to directly model all types of behaviors without considering the differences between behaviors. Zhang *et al.* [200] constructs all user behaviors in one graph and performs graph convolution operations. Wang [150, 151] extracts each behavior from the graph to construct a subgraph, then learns from the subgraph, and finally aggregates through the gating mechanism. Chen *et al.* [16] proposed a intuitive method which assigns a representation to users, items, and behaviors. In the process of propagation, the representations of edges and neighbor nodes need to be composited first to obtain a new node representation, and then it can be applied to the GNN method. Through the composition operation, the representation of the node is fused with different types of behaviors. Xia *et al.* [172] also redesign the aggregation mechanism on the graph convolutional network to explicitly model

the impact of different types of behavior. Jin *et al.* [67] assign different learnable weights to different edges to model the importance of the behaviors. In addition, in order to capture complex multi-behavior relationships, there are also some works that rely on knowledge. For example, Xia *et al.* [173] learns the representations in different behavior spaces, and then inject temporal contextual information into the representations to model the user's behavior dynamics, and finally, through the attention mechanism, discriminate the most important relationships and behaviors for the predicted target. Xia *et al.* [174] uses meta-graph networks to learn meta knowledge for different behaviors and then transfer the learned meta-knowledge between different types of behaviors.

In addition to modeling the influence of different types of behaviors, different behaviors may contain different meanings or semantics. For example, for items added to a shopping cart, users may have similar preferences for them, or these items have a complementary relationship and generally need to be purchased at the same time. If these items are connected through a graph, the representation of the item can be enhanced. In order to get better item representation, Yu *et al.* [189] not only connects related items in the graph, but also constructs a new graph of the category that the items belong to, which is used to enhance the representation of the item. It needs to construct the item graph separately, and there are also some works that do not construct the graph separately and directly in the user-item heterogeneous graph, by using for meta-path or second-order neighbors, where similar items are aggregated to enhance their representation [67, 173]. These works' details are presented in Table 12.

4.4 GNN for Different Recommendation Objectives

4.4.1 GNN for Diversity. With respect to individual-level diversity, retrieved items from recommender systems need to cover more topics such as different categories of products or different genres of music. Therefore, utilizing GNN to increase diversity requires that the learned user embeddings should be close to item embeddings with various topics. However, as the embedding aggregation operation in GNN makes user embeddings close to embeddings of items that are interacted in historical records, GNN might discourage diversity by recommending too many similar items that belong to the dominant topic in users' interaction history. For example, the learned embedding from GNN for a user who mainly interacts with electronics may be too close to embeddings of electronic items, making the GNN only recommend electronics to this user, which leads to low diversity. Therefore, in order to overcome the first challenge of weak signals from disadvantaged topics, efforts have been made to restrict the importance of dominant topics (electronics in the above example) by constructing diversified sub-graphs from the original user-item bipartite graph. Specifically, Sun *et al.* [136] propose a model called Bayesian Graph Collaborative Filtering (BGCF), which constructs augmented graphs with node copying [115] from high-order neighbors, such that items of diverse topics with high similarity can be directly connected to user nodes. Zheng *et al.* [205] propose Diversified Graph Convolutional Networks (DGCN) and conduct rebalanced neighbor sampling, which down-weights dominant topics and boosts the importance of disadvantaged topics in neighbor nodes. Fig. 18 illustrates the comparison between BGCF and DGCN on graph construction for diversity. Meanwhile, to address the second challenge of balancing between accuracy and diversity, BGCF re-ranks the top items according to item popularity, and DGCN utilizes adversarial learning on the item embeddings which encourages GNN to capture user preferences that are largely independent with item categories, which further improves recommendation diversity.

As for system-level diversity, the main target is to discover more relevant items from the long-tail ones, which have much fewer training samples compared with those popular items. To address the weak signals of long-tail items, Li *et al.* [82] propose a model called V2HT to construct an item graph that explores item correlations with external knowledge. Specifically, four types of edges are introduced, which connect frequent items and long-tail items. Then multiple GCN layers are
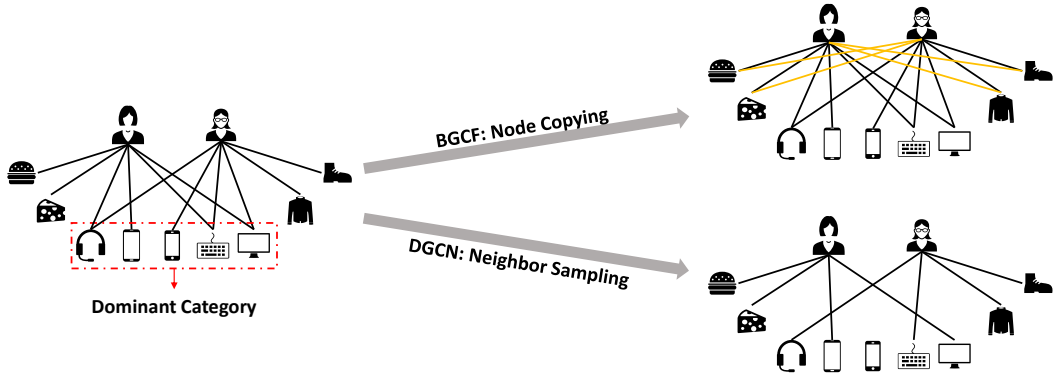
Fig. 18. Illustration of BGCF [136] and DGCN [205] on addressing the challenge of weak preference signals.

Table 13. Details of GNN models for diversified recommendation (how they address challenges)

Model	C1: Weak Preference Signals	C2: Accuracy-Diversity Balance
BGCF [136]	Node Copying	Re-rank
DGCN [205]	Neighbor Sampling	Adversarial training
V2HT [82]	Utilize item correlations	-
FH-HAT [180]	Heterogeneous graph	Diversity loss
[66]	NN & FN graph	Joint training

stacked, which propagates well-trained embeddings of frequent items to undertrained embeddings of long-tail items. In this way, long-tail item embeddings of higher quality are obtained since they share the information from frequent items; thus system-level diversity is improved with more recommendation on long-tail items.

In addition, a few studies [66, 180] utilize GNN to improve both individual-level and system-level diversity. Specifically, Xie *et al.* [180] propose FH-GAT, which addresses the challenge of weak signals by constructing a heterogeneous interaction graph to express diverse user preferences. A neighbor similarity-based loss is conducted on the heterogeneous graph to balance between accuracy and diversity. Isufi *et al.* [66] propose two GCN on the nearest neighbor (NN) graph and the furthest neighbor (FN) graph, where NN guarantees accuracy and FN enhances the weak signals of diverse items. Meanwhile, the two GCN are jointly optimized with a hyper-parameter to achieves a trade-off between accuracy and diversity. Table 13 shows the differences among the above approaches.

4.4.2 GNN for Explainablility. With the proliferation of GNN, researchers also make endeavors to improve the explainability of recommender systems with GNN's power of modeling logical relations. He *et al.* [54] construct a heterogeneous graph with three kinds of nodes, including the user, the item, and the aspect (the specific item property extracted from textual reviews). Therefore, they cast the recommendation task into a ternary relation ranking task, and propose TriRank with a high degree of explainability by explicitly modeling aspects in reviews.

Inspired by this work, the following research further explores rich information in dimensions of users and items, generally organized in a form of knowledge graph, in order to enhance the explainability [201]. Ai *et al.* [5] construct a knowledge graph with entities, *i.e.,* users and items, and relations, *e.g.,* "User A *purchased* Item B *belonging to* Category C". Moreover, they embed each entity for recommendation and adopt the shortest relation path between a user and an item in the

KGAT [KDD'19]
HAGERec [KBS'20]

Learn KG with GAT:
• model user-item interaction in a static fashion.

Overcome shortcomings of meta-path solutions:
• Discovering meta-paths

RuleRec [WWW'19]
PGPR [SIGIR'19]

EIUM[MM'19]
KPRN[AAAI'19]
TMER [WADM'21]

KG-enhanced explainability in SR:
• Combine sequential information with meta-paths in KG

Enhance explainability with KG:
• Depend on meta-paths between a user and an item
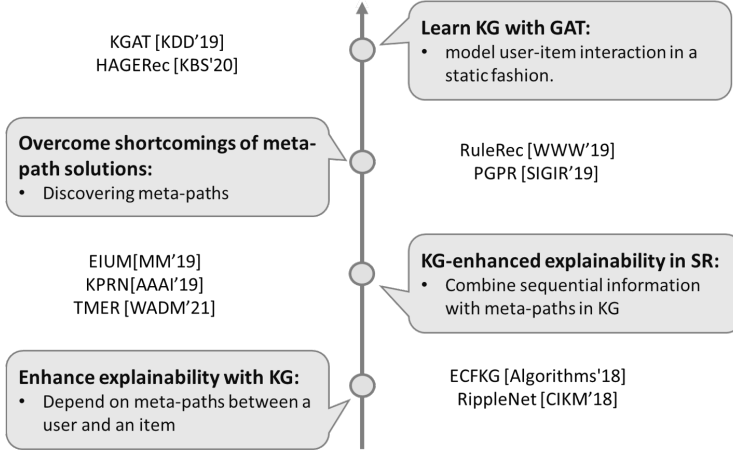
ECFKG [Algorithms'18]
RippleNet [CIKM'18]

Fig. 19. Illusration of GNN models for explainable recommendation

Table 14. Details of GNN models for explainable recommendation (how they achieve explainable recommendation)

Model	GNN	Source of Explainability
ECFKG [5]	Meta-path based GE	Meta-path over knowledge graph
RippleNet [147]	Meta-path based GraphSage	Meta-path over knowledge graph
EIUM [65]	Meta-path based GE	Meta-path over knowledge graph
KPRN [155]	Meta-path based GE	Meta-path over knowledge graph
TMER [20]	Meta-path based GE	Meta-path over knowledge graph and temporal dependency
RuleRec [101]	Meta-path based GE	Discovered meta-path over knowledge graph
PGPR [177]	Meta-path based GE	Discovered meta-path over knowledge graph
KGAT [152]	GAT	Attention mechanism
HAGERec [185]	GAT	Attention mechanism

knowledge graph to indicate the recommendation explanations. Different from separately utilizing embedding-based and path-based methods like Ai *et al.* [5], Wang *et al.* [147] proposed an end-to-end framework RippleNet which combines the two knowledge graph-aware recommendation methods together. Here, the knowledge graph contains the related knowledge of the recommended items, for example, the type and author of a movie. In this way, explanations can be generated by the path between users' history to an item with high scores.

The meta-path-based utilization of knowledge graph can also benefit other specific recommendation tasks, e.g., sequential recommendation [65, 155]. Huang *et al.* [65] extract semantic meta-paths between a user and an item from knowledge graph to help sequential recommendation. Further, they encode and rank the meta-paths to generate the recommendation list, and these meta-paths also indicate the respective explanation. Similarly, Wang *et al.* [155] also leverages knowledge graph to improve the performance of the sequential recommendation task and encode the meta-path between a user and an item with recurrent neural networks. Chen *et al.* [20] further model the temporal meta-paths by capturing historical item features and path-defined context with neural networks.

However, these meta-path-based solutions also face some challenges in terms of how to obtain these meta-paths [101, 177]. First, considering predefined meta-paths requests extensive domain
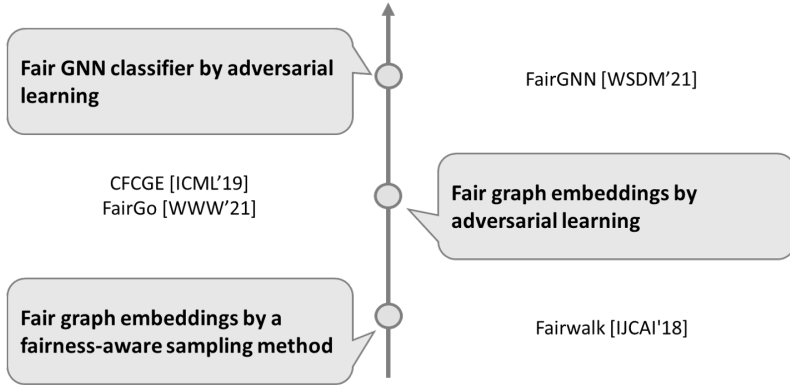
Fig. 20. Illustration of GNN models for fair recommendation.

Table 15. Details of GNN models for fair recommendation.

Model	GNN/GE	Methods Enhancing Fairness
Fairwalk [124]	Node2vec (GE)	learning fair graph embeddings by sampling
CFCGE [10]	Invariant GE	learning fair graph embeddings by adversarial learning
FairGo [162]	HetGNN(GNN)	learning fair graph embeddings by adversarial learning
FairGNN [31]	GraphSage(GNN)	learning fair GNN classifiers by adversarial learning

knowledge, Ma *et al.* [101] jointly combine the discovery of inductive rules (meta-paths) from the item-centric knowledge graph, which equips the framework with explainability and the learning of a rule-guided recommendation model. Moreover, to overcome the computational difficulties of enumerating all potential meta-paths, Xian *et al.* [177] replace the enumeration method with the reinforcement reasoning approach to identify proper meta-paths for scalability.

Besides meta-path-based solutions, Wang *et al.* [152] propose a new method named Knowledge Graph Attention Network (KGAT), where the attention mechanism can offer explainability to some extent. Yang *et al.* [185] develop a hierarchical attention graph convolutional network to model higher-order relations in the heterogeneous knowledge graph, where the explainability is also dependent on the attention mechanism.

4.4.3 GNN for Fairness. Despite the power of graph data in recommendation, it might inherit or even amplify discrimination and the societal bias in recommendation [31, 124, 162]. Past research has proven that compared with models that only adopt node attributes, the user unfairness is magnified due to the utilization of graph structures [31].

To curb the fairness issue in recommendation, some researchers propose to learn fair graph embeddings [10, 124, 162]. Rahman *et al.* [124] extends the well-known graph embedding method, node2vec [47], to a more fair version, Fairwalk, which can generate more diverse network neighborhood representation for social recommendations by sampling the next node based on its sensitive attributes. Thus, all nodes' sensitive attributes are indispensable. Bose *et al.* [10] proposes an adversarial framework to minimize the sensitive information in graph embeddings with a discriminator enforcing the fairness constraints. Moreover, considering the evaluations of fairness can be varying, fairness constraints are flexible according to the task. However, similar to Fairwalk [124], all nodes' sensitive attributes are required. Wu *et al.* [162] learn fair embeddings for recommendation from a graph-based perspective. They propose FairGo, which adopts a graph-based adversarial learning

method to map embeddings from any recommendation models into a sensitive-information-filtered space, therefore eliminating potential leakage of sensitive information from both original recommendation embeddings and user-centric graph structures.

Indeed, except for learning fair embeddings, Dai *et al.* [31] propose FairGNN, which learns fair GNN classifiers with limited known sensitive attributes in an adversarial learning paradigm with fairness constraints. Different from fair graph embeddings, fair GNN classifiers are to ensure node classification task (rather than graph embeddings) independent with sensitive data. Moreover, they develop GNN-based sensitive data estimators to overcome the issue of missing sensitive data in the real world.

4.5 GNN for Specific Recommendation Applications

Graph neural networks are also widely used to handle the specific challenges in different applications of recommender systems. As for e-commerce/product recommendation, most of the existing works have been introduced in the above sections. Li *et al.* [88] propose to stack multiple GNN modules and use a deterministic clustering algorithm to help improve the efficiency of GNN in large-scale e-commerce applications. Liu *et al.* [95] propose to leverage the topology of item-relations for building graph neural networks in e-commerce recommendation. As for the Point-of-Interest recommendation, GGLR [12] uses the sequence of POIs a user visited to construct the POI-POI graph, where the edges between POIs denote the frequency of users consecutively visiting two POIs. Zhang *et al.* [197] propose to combine social networks and user-item interactions together, which deploys embedding aggregation on both social-connected users and interacted POIs. As for news recommendation, Hu *et al.* [61] propose to introduce the preference disentanglement into the user-news embedding propagation.

There are a lot of papers for specific applications, but they can be well categorized into and covered by the corresponding stages, scenarios, and objectives, and thus we omit them here.

5 OPEN PROBLEMS AND FUTURE DIRECTIONS

5.1 Go Deeper

Due to the over-smoothing problem, more and more studies focus on properly increasing GNN's layers to capture higher-order connectivity correlations on graphs as well as improve models' performance [19, 76, 127, 213]. Despite these advancements, there is still no universal solution for constructing very deep GNN like CNN, and relevant works propose different strategies. Lai *et al.* [76] develops a meta-policy to adaptively select the number of propagation for each graph node through training with reinforcement learning (RL). The experiment results show that partial nodes require more than three layers of propagation to boost model performance. Rong *et al.* [127] alleviates the over-smoothing problem by randomly removing graph edges, which acts as a message-passing reducer. Claudio *et al.* [42] considers GNN as a dynamic system, and learned representations are the system's fixed points. Following this assumption, the transformation matrix in propagation is first fixed under stability conditions. Furthermore, only embeddings are updated in the learning procedure, leaving the matrix untrained. In this way, GNN can be trained faster as well as go deeper. Li *et al.* [80] transfer the concepts of residual/dense connections and dilated convolutions from CNNs to assist deeper GNNs. As for future works, the performance leap compared with current shallow GNNs should be an essential problem in developing very deep GNNs, like groundbreaking works in the area of CNN [64, 141]. At the same time, the computation and time complexity must also be acceptable.

5.2 Dynamic GNN in Recommendation

Existing GNN-based recommendation models are almost based on the static graph, as mentioned above, while there is plenty of *dynamics* in recommender systems. For example, in the sequential recommendation or session-based recommendation, the users' data is collected in a dynamic manner, naturally. In addition, modeling the dynamic user preferences is one of the most critical challenges in these recommendation scenarios. In addition, the platform may dynamically involve new users, new products, new features, etc., which poses challenges to static graph neural networks. Recently, dynamic graph neural networks [81, 103] have attracted attention, which deploys graph neural networks-based models on dynamically constructed graphs. Given the time-evolving property of recommender systems, the dynamic graph neural network-based recommendation model will be a promising research direction, with broad applications in the real world.

5.3 KG-enhanced recommendation with GNN

Recent research has demonstrated the power of KG in recommendation by enriching the user-item bipartite graph with knowledge [44]. Specifically, the utilization of knowledge graph with GNN significantly addresses some practical issues in recommender systems [44], for example, cold start problem [34, 198], scalability [183], and dynamicity [140]. Moreover, knowledge graph also offers a novel solution to some scenarios in recommendations, e.g., sequential recommendation [65, 155], and objectives, e.g., explainability [152, 185]. Currently, the majority of research first uses GNN to learn embeddings of knowledge graphs and then incorporate these embeddings into the recommendation model, so that an end-to-end model can be trained [145]. Therefore, we point out the utilization of KG in recommendation with GNN can be further enhanced from perspectives of data, scenario, and model. Specifically, the incorporated KG mostly records the rich item-item relations, e.g., Movie A belongs to Category B [44], but user-user knowledge is lacking in formal and plausible definition and thus substantially overlooked. Future work can consider creating user-centric KG on the foundation of abundant knowledge in sociology. Moreover, considering its success in scenarios, such as sequential recommendation [65, 155], it is promising that the leverage of KG could further enhance recommendation quality from the aspects that require more external knowledge, such as diversity and fairness. Further, existing methods on leveraging KG in recommendation cannot fully model complex relations between a user and a specific item or its attributes. Thus, designing a better framework to carve out these complex relations is another future direction.

5.4 Efficiency and Scalability

Early works on GNN follow the full-batch gradient descent algorithm, where the whole adjacency matrix is multiplied on the node embeddings during each inference step, which can not handle real-world recommender systems since the number of nodes and edges can reach a million-level scale. Hamilton *et al.* propose GraphSAGE [52] which performs neighbor sampling and only updates the related sub-graph instead of the whole graph during each inference step. The sampling strategy is also adopted in a few other works [21, 27] which reduce the computation complexity of GNN and improve the scalability. Ying *et al.* [188] successfully apply GraphSAGE to web-scale recommender systems, which can efficiently compute embeddings for billions of items. In addition, a few open-source tools have been released which can accelerate the research and development of GNN-based recommendation, such as PyG [39], DGL [149] and AliGraph [214]. We refer to another survey [1] for details on computing and accelerating GNN. Despite these existing approaches, how to achieve large-scale GNN based recommendation is still a challenging task, especially in the ranking stage

where thousands of features are involved, which results in a large and complicated heterogeneous graph.

5.5 Self-supervised GNN

The direct supervision from interaction data is relatively sparse compared with the scale of the graph. Therefore, it is necessary to include more supervision signals from the graph structure itself or the recommendation task. For example, Yu *et al.* [194] and Wu *et al.* [161] attempt to enhance GNN-based recommendation by designing auxiliary tasks from the graph structure with self-supervision. Data augmentations such as node dropout are utilized to generate sample pairs for contrastive training. We believe that it is a promising future direction to leverage extra self-supervised tasks to learn meaningful and robust representations for GNN-based recommender systems.

5.6 Conversational Recommendation with GNN

In existing recommender systems, there may exist the issue of information asymmetry that the system can only estimate users' preferences based on their *historically collected* behavior data. To address it, recently, conversational (interactive) recommendation researches [77, 139], propose the new paradigm the user can interact with the system in conversations, and then new data can be dynamically collected. Specifically, users can chat with the system to explicitly convey their consumption demands or offer positive/negative feedback on the recommended items. As future work, the advances of representation learning with graph neural networks can be combined with preference learning in the conversational recommendation.

5.7 AutoML-enhanced GNN

Recommendation scenarios are diverse and largely different from each other; thus, there exists no silver bullet GNN model that can generalize across all scenarios. Recently, AutoML (Automated Machine Learning) [187] is proposed, which can automatically design appropriate models for specific tasks. With respect to GNN-based recommendation, the search space is quite large, which includes multiple options for the neighbor sampler, aggregator, interaction function, and so on. As a consequence, AutoML can, to a great extent, reduce human effort in discovering advanced model structures. A few works [45, 62] have been proposed which search to combine GNN layers and aggregate neighbors. However, designing AutoML algorithms to search for GNN based recommender systems is a largely unexplored yet promising future direction.

6 CONCLUSION

There is a rapid development of graph neural networks models in the research field of recommender systems. This paper provides an extensive survey systematically presenting the challenges, methods, and future directions in this area. Not only the history of development and also the most recent advances are well covered and introduced. We hope this survey can well help both junior and experienced researchers in the relative areas.

REFERENCES

[1] Sergi Abadal, Akshay Jain, Robert Guirado, Jorge López-Alonso, and Eduard Alarcón. 2020. Computing graph neural networks: A survey from algorithms to accelerators. *arXiv preprint arXiv:2010.00130* (2020).

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 42–46.

[3] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286* (2019).

[4] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2020. The Connection Between Popularity Bias, Calibration, and Fairness in Recommendation. In *Fourteenth ACM Conference on Recommender Systems*. Association for Computing Machinery, New York, NY, USA, 726–731.

[5] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* 11, 9 (2018), 137.

[6] Ting Bai, Youjie Zhang, Bin Wu, and Jian-Yun Nie. 2020. Temporal Graph Neural Networks for Social Recommendation. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 898–903.

[7] Robert M Bell and Yehuda Koren. 2007. Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter* 9, 2 (2007), 75–79.

[8] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2018. Graph convolutional matrix completion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[9] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in Recommendation Ranking through Pairwise Comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, New York, NY, USA, 2212–2220.

[10] Avishek Bose and William Hamilton. 2019. Compositional fairness constraints for graph embeddings. In *International Conference on Machine Learning*. PMLR, 715–724.

[11] Da Cao, Liqiang Nie, Xiangnan He, Xiaochi Wei, Shunzhi Zhu, and Tat-Seng Chua. 2017. Embedding factorization models for jointly recommending items and user generated lists. In *SIGIR*. 585–594.

[12] Buru Chang, Gwanghoon Jang, Seoyoon Kim, and Jaewoo Kang. 2020. Learning graph-based geographical latent representation for point-of-interest recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 135–144.

[13] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Bundle recommendation with graph convolutional networks. In *Proceedings of the 43rd international ACM SIGIR conference on Research and development in Information Retrieval*. 1673–1676.

[14] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential Recommendation with Graph Neural Networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 378–387.

[15] Chen Chen, Jie Guo, and Bin Song. 2021. Dual Attention Transfer in Session-based Recommendation with Multi-dimensional Integration. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 869–878.

[16] Chong Chen, Weizhi Ma, Min Zhang, Zhaowei Wang, Xiuqiang He, Chenyang Wang, Yiqun Liu, and Shaoping Ma. 2021. Graph Heterogeneous Multi-Relational Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 3958–3966.

[17] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 225–234.

[18] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3438–3445.

[19] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3438–3445.

[20] Hongxu Chen, Yicong Li, Xiangguo Sun, Guandong Xu, and Hongzhi Yin. 2021. Temporal Meta-Path Guided Explainable Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 1056–1064.

[21] Jie Chen, Tengfei Ma, and Cao Xiao. 2018. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247* (2018).

[22] Liang Chen, Yang Liu, Xiangnan He, Lianli Gao, and Zibin Zheng. 2019. Matching user with item set: collaborative bundle recommendation with deep attention network. In *IJCAI*. 2095–2101.

[23] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1172–1180.

[24] Tianwen Chen and Raymond Chi-Wing Wong. 2021. An Efficient and Effective Framework for Session-based Social Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 400–408.

[25] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Twenty-Third international joint conference on Artificial Intelligence*.

[26] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[27] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 257–266.

[28] Robert B Cialdini and Noah J Goldstein. 2004. Social influence: Compliance and conformity. *Annu. Rev. Psychol.* 55 (2004), 591–621.

[29] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[30] Qiang Cui, Tao Wei, Yafeng Zhang, and Qing Zhang. 2020. HeroGRAPH: A Heterogeneous Graph Framework for Multi-Target Cross-Domain Recommendation.. In *ORSUM@ RecSys*.

[31] Enyan Dai and Suhang Wang. 2021. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information *(WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 680–688.

[32] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*. 293–296.

[33] Qilin Deng, Kai Wang, Minghao Zhao, Zhene Zou, Runze Wu, Jianrong Tao, Changjie Fan, and Liang Chen. 2020. Personalized Bundle Recommendation in Online Games. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2381–2388.

[34] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-Guided Heterogeneous Graph Neural Network for Intent Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &; Data Mining (KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2478–2486.

[35] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference*. 417–426.

[36] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. 2021. Continuous-Time Sequential Recommendation with Temporal Graph Collaborative Transformer. *arXiv preprint arXiv:2108.06625* (2021).

[37] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new poi recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[38] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3558–3565.

[39] Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428* (2019).

[40] Alex M Fout. 2017. *Protein interface prediction using graph convolutional networks*. Ph.D. Dissertation. Colorado State University.

[41] Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. 2019. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 94–101.

[42] Claudio Gallicchio and Alessio Micheli. 2020. Fast and deep graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3898–3905.

[43] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. 2019. Cross-domain recommendation without sharing user-relevant data. In *The world wide web conference*. 491–502.

[44] Yang Gao, Yi-Fan Li, Yu Lin, Hang Gao, and Latifur Khan. 2020. Deep learning on knowledge graph for recommender system: A survey. *arXiv preprint arXiv:2004.00387* (2020).

[45] Yang Gao, Hong Yang, Peng Zhang, Chuan Zhou, and Yue Hu. 2019. Graphnas: Graph neural architecture search with reinforcement learning. *arXiv preprint arXiv:1904.09981* (2019).

[46] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2221–2231.

[47] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[48] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[49] Lei Guo, Li Tang, Tong Chen, Lei Zhu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2021. DA-GCN: A Domain-aware Attentive Graph Convolution Network for Shared-account Cross-domain Sequential Recommendation. *arXiv preprint arXiv:2105.03300* (2021).

[50] Wei Guo, Rong Su, Renhao Tan, Huifeng Guo, Yingxue Zhang, Zhirong Liu, Ruiming Tang, and Xiuqiang He. 2021. Dual Graph enhanced Embedding Neural Network for CTRPrediction. *arXiv preprint arXiv:2106.00314* (2021).

[51] Zhiwei Guo and Heng Wang. 2020. A deep graph neural network-based mechanism for social recommendations. *IEEE Transactions on Industrial Informatics* 17, 4 (2020), 2776–2783.

[52] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 1025–1035.

[53] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis* 30, 2 (2011), 129–150.

[54] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1661–1670.

[55] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[56] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[57] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[58] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[59] Cheng Hsu and Cheng-Te Li. 2021. RetaGNN: Relational Temporal Attentive Graph Neural Networks for Holistic Sequential Recommendation. In *Proceedings of the Web Conference 2021*. 2968–2979.

[60] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 667–676.

[61] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph neural news recommendation with unsupervised preference disentanglement. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4255–4264.

[62] ZHAO Huan, YAO Quanming, and TU Weiwei. 2021. Search to aggregate neighborhood for graph neural network. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 552–563.

[63] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. 2021. Knowledge-aware coupled graph neural network for social recommendation. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[64] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[65] Xiaowen Huang, Quan Fang, Shengsheng Qian, Jitao Sang, Yan Li, and Changsheng Xu. 2019. Explainable Interaction-Driven User Modeling over Knowledge Graph for Sequential Recommendation. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*. Association for Computing Machinery, New York, NY, USA, 548–556.

[66] Elvin Isufi, Matteo Pocchiari, and Alan Hanjalic. 2021. Accuracy-diversity trade-off in recommender systems via graph convolutions. *Information Processing & Management* 58, 2 (2021), 102459.

[67] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 659–668.

[68] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1563–1572.

[69] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[70] Wang-Cheng Kang and Julian McAuley. 2019. Candidate generation with binary codes for large-scale top-n recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1523–1532.

[71] Kyung-Min Kim, Donghyun Kwak, Hanock Kwak, Young-Jin Park, Sangkwon Sim, Jae-Han Cho, Minkyu Kim, Jihun Kwon, Nako Sung, and Jung-Woo Ha. 2019. Tripartite heterogeneous graph propagation for large-scale social recommendation. *arXiv preprint arXiv:1908.02569* (2019).

[72] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

[73] Yehuda Koren. 2009. The bellkor solution to the netflix grand prize. *Netflix prize documentation* 81, 2009 (2009), 1–10.

[74] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[75] Johannes Kunkel, Tim Donkers, Lisa Michael, Catalin-Mihai Barbu, and Jürgen Ziegler. 2019. Let me explain: impact of personal and impersonal explanations on trust in recommender systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12.

[76] Kwei-Herng Lai, Daochen Zha, Kaixiong Zhou, and Xia Hu. 2020. Policy-gnn: Aggregation optimization for graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 461–471.

[77] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2073–2083.

[78] Jurek Leonhardt, Avishek Anand, and Megha Khosla. 2018. User fairness in recommender systems. In *Companion Proceedings of the The Web Conference 2018*. 101–102.

[79] Chen Li, Linmei Hu, Chuan Shi, Guojie Song, and Yuanfu Lu. 2021. Sequence-aware Heterogeneous Graph Neural Collaborative Filtering. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 64–72.

[80] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2019. Deepgcns: Can gcns go as deep as cnns?. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9267–9276.

[81] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. 2020. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 214–223.

[82] Mengmeng Li, Tian Gan, Meng Liu, Zhiyong Cheng, Jianhua Yin, and Liqiang Nie. 2019. Long-tail hashtag recommendation for micro-videos with graph convolutional network. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 509–518.

[83] Xingchen Li, Xiang Wang, Xiangnan He, Long Chen, Jun Xiao, and Tat-Seng Chua. 2020. Hierarchical fashion graph network for personalized outfit recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 159–168.

[84] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented Fairness in Recommendation. In *Proceedings of the Web Conference 2021*. 624–632.

[85] Yang Li, Tong Chen, Hongzhi Yin, and Zi Huang. 2021. Discovering Collaborative Signals for Next POI Recommendation with Iterative Seq2Graph Augmentation. *arXiv preprint arXiv:2106.15814* (2021).

[86] Yunqi Li, Yingqiang Ge, and Yongfeng Zhang. 2021. Tutorial on Fairness of Machine Learning in Recommender Systems. SIGIR.

[87] Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. Fi-gnn: Modeling feature interactions via graph neural networks for ctr prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 539–548.

[88] Zhao Li, Xin Shen, Yuhang Jiao, Xuming Pan, Pengcheng Zou, Xianling Meng, Chengwei Yao, and Jiajun Bu. 2020. Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1677–1688.

[89] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.

[90] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Xueou Wang, Yong Liang Goh, Renrong Weng, and Jagannadan Varadarajan. 2020. STP-UDGAT: Spatial-Temporal-Preference User Dimensional Graph Attention Network for Next POI Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 845–854.

[91] Feng Liu, Weiwen Liu, Xutao Li, and Yunming Ye. 2020. Inter-sequence Enhanced Framework for Personalized Sequential Recommendation. *arXiv preprint arXiv:2004.12118* (2020).

[92] Meng Liu, Jianjun Li, Guohui Li, and Peng Pan. 2020. Cross Domain Recommendation via Bi-directional Transfer Graph Collaborative Filtering Networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 885–894.

[93] Qi Liu, Ruobing Xie, Lei Chen, Shukai Liu, Ke Tu, Peng Cui, Bo Zhang, and Leyu Lin. 2020. Graph neural network for tag ranking in tag-enhanced video recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2613–2620.

[94] Weiwen Liu, Qing Liu, Ruiming Tang, Junyang Chen, Xiuqiang He, and Pheng Ann Heng. 2020. Personalized Re-ranking with Item Relationships for E-commerce. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 925–934.

[95] Weiwen Liu, Yin Zhang, Jianling Wang, Yun He, James Caverlee, Patrick PK Chan, Daniel S Yeung, and Pheng-Ann Heng. 2021. Item Relationship Graph Neural Networks for E-Commerce. *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[96] Yang Liu, Chen Liang, Xiangnan He, Jiaying Peng, Zibin Zheng, and Jie Tang. 2020. Modelling high-order social relations for item recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[97] Yidan Liu, Min Xie, and Laks VS Lakshmanan. 2014. Recommending user generated item lists. In *RecSys*. 185–192.

[98] Dongsheng Luo, Yuchen Bian, Xiang Zhang, and Jun Huan. 2020. Attentive Social Recommendation: Towards User And Item Diversities. *arXiv preprint arXiv:2011.04797* (2020).

[99] Xusheng Luo, Luxin Liu, Yonghua Yang, Le Bo, Yuanpeng Cao, Jinghang Wu, Qiang Li, Keping Yang, and Kenny Q Zhu. 2020. AliCoCo: Alibaba e-commerce cognitive concept net. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 313–327.

[100] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory augmented graph neural networks for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5045–5052.

[101] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly Learning Explainable Rules for Recommendation with Knowledge Graph. In *The World Wide Web Conference (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 1210–1221.

[102] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[103] Yao Ma, Ziyi Guo, Zhaocun Ren, Jiliang Tang, and Dawei Yin. 2020. Streaming graph neural networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 719–728.

[104] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach.. In *IJCAI*, Vol. 17. 2464–2470.

[105] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. FairMatch: A Graph-Based Approach for Improving Aggregate Diversity in Recommender Systems. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '20)*. Association for Computing Machinery, New York, NY, USA, 154–162. https://doi.org/10.1145/3340631.3394860

[106] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. 165–172.

[107] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[108] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a Fair Marketplace: Counterfactual Evaluation of the Trade-off between Relevance, Fairness & Satisfaction in Recommendation Systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, New York, NY, USA, 2243–2251.

[109] Wenjing Meng, Deqing Yang, and Yanghua Xiao. 2020. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1091–1100.

[110] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*.

[111] Weiqing Min, Shuqiang Jiang, and Ramesh Jain. 2019. Food Recommendation: Framework, Existing Solutions, and Challenges. *IEEE Transactions on Multimedia* 22, 10 (2019), 2659–2671.

[112] Nima Mirbakhsh and Charles X Ling. 2015. Improving top-n recommendation for cold-start users via cross-domain information. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 9, 4 (2015), 1–19.

[113] Nan Mu, Daren Zha, Yuanye He, and Zhihao Tang. 2019. Graph attention networks for neural social recommendation. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 1320–1327.

[114] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1933–1942.

[115] Soumyasundar Pal, Florence Regol, and Mark Coates. 2019. Bayesian graph convolutional neural networks using node copying. *arXiv preprint arXiv:1911.04965* (2019).

[116] Zhiqiang Pan, Fei Cai, Wanyu Chen, Honghui Chen, and Maarten de Rijke. 2020. Star graph neural networks for session-based recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge*

Management. 1195–1204.

[117] Ioannis Paparrizos, B Barla Cambazoglu, and Aristides Gionis. 2011. Machine learned job recommendation. In *Proceedings of the fifth ACM Conference on Recommender Systems.* 325–328.

[118] Apurva Pathak, Kshitiz Gupta, and Julian McAuley. 2017. Generating and personalizing bundle recommendations on steam. In *SIGIR.* 1073–1076.

[119] Georgina Peake and Jun Wang. 2018. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2060–2069.

[120] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, et al. 2019. Personalized re-ranking for recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems.* 3–11.

[121] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* 701–710.

[122] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 579–588.

[123] Ruihong Qiu, Hongzhi Yin, Zi Huang, and Tong Chen. 2020. Gag: Global attributed graph neural network for streaming session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 669–678.

[124] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19.* International Joint Conferences on Artificial Intelligence Organization, 3289–3295. https://doi.org/10.24963/ijcai.2019/456

[125] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining.* IEEE, 995–1000.

[126] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence.* 452–461.

[127] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *International Conference on Learning Representations.*

[128] Heng-Shiou Sheu and Sheng Li. 2020. Context-aware graph embedding for session-based news recommendation. In *Fourteenth ACM conference on recommender systems.* 657–662.

[129] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine* 30, 3 (2013), 83–98.

[130] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18).* Association for Computing Machinery, New York, NY, USA, 2219–2228.

[131] Rashmi Sinha and Kirsten Swearingen. 2002. The role of transparency in recommender systems. In *CHI'02 extended abstracts on Human factors in computing systems.* Association for Computing Machinery, New York, NY, USA, 830–831.

[132] Changhao Song, Bo Wang, Qinxue Jiang, Yehua Zhang, Ruifang He, and Yuexian Hou. 2021. Social Recommendation with Implicit Social Influence. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1788–1792.

[133] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 1161–1170.

[134] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM international conference on web search and data mining.* 555–563.

[135] Yixin Su, Rui Zhang, Sarah Erfani, and Zhenghua Xu. 2021. Detecting Beneficial Feature Interactions for Recommender Systems. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI).*

[136] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huifeng Guo, Ruiming Tang, Han Yuan, Xiuqiang He, et al. 2020. A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2030–2039.

[137] Jianing Sun, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, Xiuqiang He, Chen Ma, and Mark Coates. 2020. Neighbor interaction aware graph convolution networks for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1289–1298.

[138] Peijie Sun, Le Wu, and Meng Wang. 2018. Attentive recurrent social recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 185–194.

[139] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st international acm sigir conference on research & development in information retrieval*. 235–244.

[140] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent Knowledge Graph Embedding for Effective Recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 297–305.

[141] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[142] Nava Tintarev and Judith Masthoff. 2007. A survey of explanations in recommender systems. In *2007 IEEE 23rd international conference on data engineering workshop*. IEEE, 801–810.

[143] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Neural Information Processing Systems Conference (NIPS 2013)*, Vol. 26. Neural Information Processing Systems Foundation (NIPS).

[144] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

[145] Baocheng Wang and Wentao Cai. 2020. Knowledge-enhanced graph neural networks for sequential recommendation. *Information* 11, 8 (2020), 388.

[146] Bei Wang, Chenrui Zhang, Hao Zhang, Xiaoqing Lyu, and Zhi Tang. 2020. Dual Autoencoder Network with Swap Reconstruction for Cold-Start Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2249–2252.

[147] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 417–426.

[148] Jianling Wang, Kaize Ding, Ziwei Zhu, and James Caverlee. 2021. Session-based Recommendation with Hypergraph Attention Networks. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 82–90.

[149] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, et al. 2019. Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs. (2019).

[150] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *Proceedings of The Web Conference 2020*. 3056–3062.

[151] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2021. Incorporating Link Prediction into Multi-Relational Item Graph Modeling for Session-based Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[152] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.

[153] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[154] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1001–1010.

[155] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5329–5336.

[156] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*. 627–636.

[157] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 169–178.

[158] Mark Wilhelm, Ajith Ramanathan, Alexander Bonomo, Sagar Jain, Ed H Chi, and Jennifer Gillenwater. 2018. Practical diversified recommendations on youtube with determinantal point processes. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2165–2173.

[159] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3597–3606.

[160] Jiancan Wu, Xiangnan He, Xiang Wang, Qifan Wang, Weijian Chen, Jianxun Lian, and Xing Xie. 2020. Graph Convolution Machine for Context-aware Recommender System. *arXiv preprint arXiv:2001.11402* (2020).

[161] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 726–735.

[162] Le Wu, Lei Chen, Pengyang Shao, Richang Hong, Xiting Wang, and Meng Wang. 2021. Learning Fair Representations for Recommendation: A Graph-based Perspective. In *Proceedings of the Web Conference 2021*. 2198–2208.

[163] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2020. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[164] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 235–244.

[165] Le Wu, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2018. Collaborative neural social recommendation. *IEEE transactions on systems, man, and cybernetics: systems* (2018).

[166] Peizhi Wu, Yi Tu, Xiaojie Yuan, Adam Jatowt, and Zhenglu Yang. 2018. Neural Framework for Joint Evolution Modeling of User Feedback and Social Links in Dynamic Social Networks.. In *IJCAI*. 1632–1638.

[167] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*. 2091–2102.

[168] Shiwen Wu, Fei Sun, Wentao Zhang, and Bin Cui. 2021. Graph neural networks in recommender systems: a survey. *arXiv preprint arXiv:2011.02260* (2021).

[169] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.

[170] Shu Wu, Mengqi Zhang, Xin Jiang, Xu Ke, and Liang Wang. 2019. Personalizing graph neural networks with attention mechanism for session-based recommendation. *arXiv preprint arXiv:1910.08887* (2019).

[171] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[172] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Mengyin Lu, and Liefeng Bo. 2021. Multi-Behavior Enhanced Recommendation with Cross-Interaction Collaborative Relation Modeling. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1931–1936.

[173] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. 2021. Knowledge-Enhanced Hierarchical Graph Transformer Network for Multi-Behavior Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4486–4493.

[174] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. 2021. Graph Meta Network for Multi-Behavior Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 757–766.

[175] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. 2021. Self-Supervised Graph Co-Training for Session-based Recommendation. *arXiv preprint arXiv:2108.10560* (2021).

[176] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4503–4511.

[177] Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 285–294.

[178] Fengtong Xiao, Lin Li, Weinan Xu, Jingyu Zhao, Xiaofeng Yang, Jun Lang, and Hao Wang. 2021. DMBGN: Deep Multi-Behavior Graph Networks for Voucher Redemption Rate Prediction. *arXiv preprint arXiv:2106.03356* (2021).

[179] Min Xie, Hongzhi Yin, Fanjiang Xu, Hao Wang, and Xiaofang Zhou. 2016. Graph-based metric embedding for next poi recommendation. In *International Conference on Web Information Systems Engineering*. Springer, 207–222.

[180] Ruobing Xie, Qi Liu, Shukai Liu, Ziwei Zhang, Peng Cui, Bo Zhang, and Leyu Lin. 2021. Improving Accuracy and Diversity in Matching of Recommendation with Diversified Preference Network. *arXiv preprint arXiv:2102.03787* (2021).

[181] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation.. In *IJCAI*, Vol. 19.

3940–3946.

[182] Fengli Xu, Jianxun Lian, Zhenyu Han, Yong Li, Yujian Xu, and Xing Xie. 2019. Relation-aware graph convolutional networks for agent-initiated social e-commerce recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 529–538.

[183] Fengli Xu, Jianxun Lian, Zhenyu Han, Yong Li, Yujian Xu, and Xing Xie. 2019. Relation-Aware Graph Convolutional Networks for Agent-Initiated Social E-Commerce Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 529–538.

[184] Huance Xu, Chao Huang, Yong Xu, Lianghao Xia, Hao Xing, and Dawei Yin. 2020. Global context enhanced social recommendation with hierarchical graph neural networks. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 701–710.

[185] Zuoxi Yang and Shoubin Dong. 2020. HAGERec: hierarchical attention graph convolutional network incorporating knowledge graph for explainable recommendation. *Knowledge-Based Systems* 204 (2020), 106194.

[186] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 7370–7377.

[187] Quanming Yao, Mengshuo Wang, Yuqiang Chen, Wenyuan Dai, Yu-Feng Li, Wei-Wei Tu, Qiang Yang, and Yang Yu. 2018. Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint arXiv:1810.13306* (2018).

[188] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.

[189] Bo Yu, Ruoqian Zhang, Wei Chen, and Junhua Fang. 2021. Graph neural network based model for multi-behavior session-based recommendation. *GeoInformatica* (2021), 1–19.

[190] Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2020. TAGNN: Target attentive graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1921–1924.

[191] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-Aware Self-Supervised Tri-Training for Recommendation. *arXiv preprint arXiv:2106.03569* (2021).

[192] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. 2020. Enhance social recommendation with adversarial graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[193] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *Proceedings of the Web Conference 2021*. 413–424.

[194] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *Proceedings of the Web Conference 2021*. 413–424.

[195] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.

[196] Jun Zhang, Chen Gao, Depeng Jin, and Yong Li. 2021. Group-Buying Recommendation for Social E-Commerce. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1536–1547.

[197] Jiyong Zhang, Xin Liu, Xiaofei Zhou, and Xiaowen Chu. 2021. Leveraging graph neural networks for point-of-interest recommendations. *Neurocomputing* 462 (2021), 1–13.

[198] Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. STAR-GCN: Stacked and Reconstructed Graph Convolutional Networks for Recommender Systems. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. AAAI Press, 4264–4270.

[199] Mengqi Zhang, Shu Wu, Xueli Yu, and Liang Wang. 2021. Dynamic Graph Neural Networks for Sequential Recommendation. *arXiv preprint arXiv:2104.07368* (2021).

[200] Weifeng Zhang, Jingwen Mao, Yi Cao, and Congfu Xu. 2020. Multiplex Graph Neural Networks for Multi-behavior Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2313–2316.

[201] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101. https://doi.org/10.1561/1500000066

[202] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 83–92.

[203] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-domain recommendation via preference propagation graphnet. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2165–2168.

[204] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-domain recommendation via preference propagation graphnet. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2165–2168.

[205] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. 2021. DGCN: Diversified Recommendation with Graph Convolutional Networks. In *Proceedings of the Web Conference 2021*. 401–412.

[206] Yu Zheng, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2021. Incorporating Price into Recommendation with Graph Convolutional Networks. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[207] Yu Zheng, Chen Gao, Xiangnan He, Yong Li, and Depeng Jin. 2020. Price-aware recommendation with graph convolutional networks. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 133–144.

[208] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. 2020. DGTN: Dual-channel Graph Transition Network for Session-based Recommendation. In *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 236–242.

[209] Zhi Zheng, Chao Wang, Tong Xu, Dazhong Shen, Penggang Qin, Baoxing Huai, Tongzhu Liu, and Enhong Chen. 2021. Drug Package Recommendation via Interaction-aware Graph Induction. In *Proceedings of the Web Conference 2021*. 1284–1295.

[210] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

[211] Huachi Zhou, Qiaoyu Tan, Xiao Huang, Kaixiong Zhou, and Xiaoling Wang. 2021. Temporal Augmented Graph Neural Networks for Session-Based Recommendations. (2021).

[212] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.

[213] Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, and Xia Hu. 2020. Towards Deeper Graph Neural Networks with Differentiable Group Normalization. *Advances in Neural Information Processing Systems* 33 (2020).

[214] Rong Zhu, Kun Zhao, Hongxia Yang, Wei Lin, Chang Zhou, Baole Ai, Yong Li, and Jingren Zhou. 2019. AliGraph: a comprehensive graph neural network platform. *Proceedings of the VLDB Endowment* 12, 12 (2019), 2094–2105.

[215] Tianyu Zhu, Leilei Sun, and Guoqing Chen. 2021. Graph-based Embedding Smoothing for Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[216] Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-aware tensor-based recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1153–1162.

[217] Tao Zhuang, Wenwu Ou, and Zhirong Wang. 2018. Globally optimized mutual influence aware ranking in e-commerce search. *arXiv preprint arXiv:1805.08524* (2018).