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

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

XIANGNAN HE, School of Information Science and Technology, University of Science and Technology of China, China

YONG LI, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, 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 to recommender systems. In this survey, we conduct a comprehensive review of the literature on 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 in this area. We summarize the representative papers along with their code repositories in https://github.com/tsinghua-fib-lab/GNN-Recommender-Systems.

CCS Concepts: • **Information systems** → **Recommender systems**; • **General and reference** → *Surveys and overviews;*

Additional Key Words and Phrases: Recommender systems, graph neural networks, graph representation learning, information retrieval

1 INTRODUCTION

Recommender system is a kind of filtering system in which the goal is to present personalized information to users, which improves the user experience and promotes business profit. As one of 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 [85, 148, 149], neural models [28, 57, 66], and GNN-based models [65, 181, 222]. The earliest recommendation models captured the **collaborative filtering (CF)** effect by directly calculating the similarity of interactions. Then model-based CF methods, such as **matrix factorization (MF)** [85] or factorization machine [148], 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 [28, 57, 66] 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)** [57] combined the shallow model **factorization machine (FM)** [148] 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 in graph neural networks provide a strong and fundamental 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 in handling the structural data and exploring 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 into graph, with nodes representing elements and edges representing relations. Second, for the specific task, the components in graph neural networks 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, and so on [218], 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 introducing a number of computations, the efficient deployment of graph neural networks in recommender systems is another critical challenge.

In this article, we aim to provide a systematic and comprehensive review of the research efforts, especially on how they improve recommendation with graph neural networks and address the corresponding challenges. To fulfill a clear understanding, we categorize research of
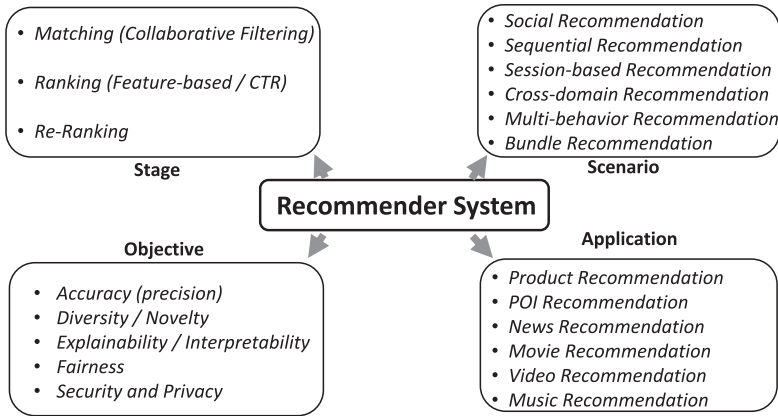
Fig. 1. An illustration of typical recommender systems (stages, scenarios, objectives, and applications).

recommender systems from four perspectives, stage, scenario, objectives, and applications, as illustrated in Figure 1. We summarize the representative papers along with their code repositories in https://github.com/tsinghua-fib-lab/GNN-Recommender-Systems.

It is worth mentioning that there is one existing survey [202] of graph neural network-based recommender system, compared with which our differences are as follows: First, it mainly considers collaborative filtering, sequential recommendation, social recommendation, and knowledge-based recommendation, which belong to different aspects based on the taxonomy and thus cannot be listed side-by-side. For example, the sequential recommendation is only one specific recommendation scenario with a special setting of input and output, as pointed out by this survey. Second, important recommendation stages (such as ranking), recommendation scenarios (such as cross-domain recommendation), and beyond-accuracy recommendation objectives are ignored. Especially, the beyond-accuracy recommender system is now a very important topic in both academia and industry. Last, since this area is developing rapidly, there are many recent papers in top-tier venues not covered by Reference [202]. It is worth mentioning that there is a recently published book chapter [32] that introduces GNN's basic concepts and several representative applications in modern recommender systems.

Furthermore, in addition to Reference [202], there are some other surveys closely relevant to this survey as follows:

- Reference [108] is a very recent survey on recommender systems based on **heterogeneous information network (HIN)**, which divides literature into three categories: similarity measurement, matrix factorization, and graph representation learning. As for the category of graph representation learning, the authors further divide it into two sub-categories: two-stage training-based methods and end-to-end training-based methods (including relation-based ones and meta-path-based ones). As for the end-to-end methods, this survey briefly introduces some graph neural network-based methods. That is, Reference [108] only covers a very small fraction of GNN-based recommenders.
- Reference [176] is a survey on graph-based recommender systems. It is very short (nine pages) and only uses one page to briefly introduce very-limited typical works of GNN-based recommenders. Reference [39] is a survey of recommender systems based on graph embedding techniques. It mainly discusses the traditional graph embedding methods and only discusses a few GNN-based methods.

- Reference [196] is a very recent survey on deep learning-based recommender systems. It mainly discusses two parts of works: deep learning-based collaborative filtering and deep learning-based feature learning for recommendation. In the first part, the authors have introduced some representative GNN-based collaborative filtering methods; in the second part, the authors introduced a few works that use GNN to extract visual features.
- There are other surveys on specific topics of recommendation, such as Reference [175] for session-based recommendation, References [30, 59] for knowledge-graph based recommendation, Reference [26] for explainable recommendation, Reference [191] for news recommendation. There are some other surveys focusing on specific techniques of recommendation, such as automated machine learning [241] and self-supervised learning [241].

In summary, our survey is largely different compared with these surveys, as our survey provides extensive and up-to-date introduction and discussions about GNN-based recommenders.

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 of 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, and so on. Besides the standard stages, there are many specific recommendation scenarios with a special definition. For example, in the past 20 years, social recommendation has been attracting attention, defined as improving recommender systems 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, and so on. As for the applications, GNN models can be widely deployed in e-commerce recommendation, point-of-interest recommendation, news recommendation, movie recommendation, music recommendation, and so on.

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 [34, 190]. 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 [34, 81]. 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

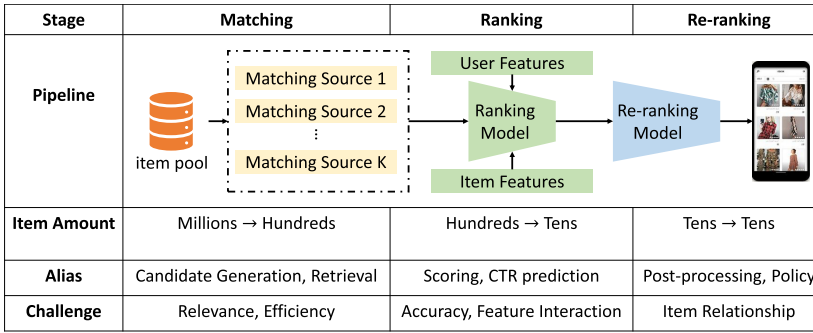| Stage	Matching	Ranking	Re-ranking
Pipeline			
Item Amount	Millions → Hundreds	Hundreds → Tens	Tens → Tens
Alias	Candidate Generation, Retrieval	Scoring, CTR prediction	Post-processing, Policy
Challenge	Relevance, Efficiency	Accuracy, Feature Interaction	Item Relationship

Fig. 2. The typical pipeline of recommender systems.

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, and so on.

- **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 [80, 101, 156]. For example, rich features including user profiles and item attributes can be taken into consideration, and advanced techniques such as self-attention [80] can be utilized. Since many 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 and so on [143]. Therefore, a re-ranking stage is necessary, which usually removes certain items or changes the order of the list to fulfill additional criteria and satisfy business needs. The main concern in this stage is to consider multiple relationships among the top-scored items [5, 255]. For example, similar or substitutable items can lead to information redundancy when they are displayed closely in the recommendations.

Figure 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 lives. 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* [33]. 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
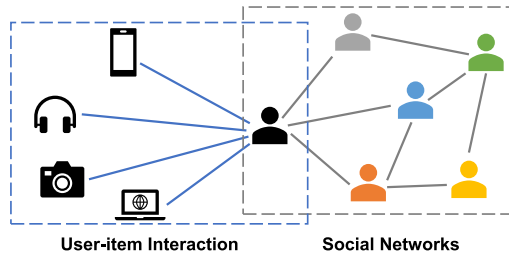
---

*http://wechat.com/.

Fig. 3. An illustration of social recommendation. User interactions are affected by both their own preferences and the social factor.
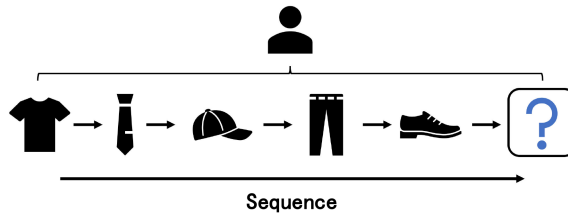

Fig. 4. An illustration of sequential recommendation. Given a user's historical sequence, recommender system aims to predict the next item.

on many social platforms, i.e., people tend to build social relations with others who have similar preferences with them [127]. 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. Figure 3 illustrates the data input of social recommendation, of which user interactions are determined by both 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 Figure 4. To symbolize this problem, for the sequence of items $\{x_1, x_2, \ldots, x_n\}$ that the user has 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 [66] 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, 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. Second, 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

Fig. 5. An illustration of session-based recommendation. Given a anonymous short session, recommender system aims to predict the next item.
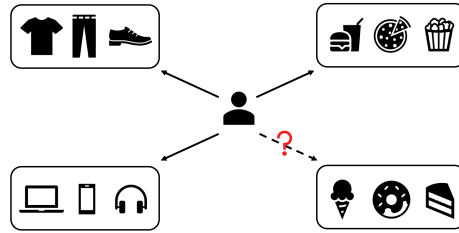
Fig. 6. An illustration of bundle recommendation.
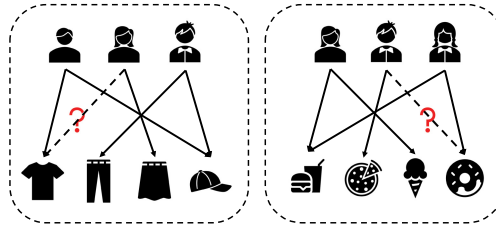
Fig. 7. An illustration of cross-domain recommendation.

track the user-ID's behaviors over a long period 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)** [31], which aims at predicting the next item with a given anonymous behavioral session data, as shown in Figure 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 [67].

- **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 [12, 23, 114, 141], as illustrated in Figure 6. Bundle recommendation is very common nowadays on online platforms, e.g., the music playlists on Spotify, the pinboards on Pinterest, the computer 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 [95] and drug packages [246].

- **Cross-domain Recommendation.** As an increasing number of users interact with multimodal information across multiple domains, **cross-domain recommendation (CDR)** has been demonstrated to be a promising method to alleviate cold start and data sparsity problems [50, 52, 70, 79, 124, 132, 238], as illustrated in Figure 7. CDR methods can be

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

roughly divided into two categories, **single-target CDR (STCDR)** and **dual-target CDR (DTCDR)** [35]. CDR methods transfer the information from the source domain to the target domain in one direction; DTCDR 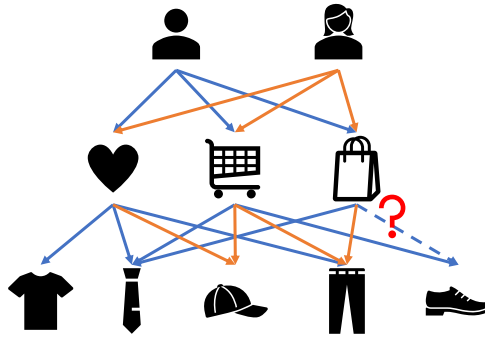emphasizes the mutual utilization of information from both the source domain and target domain, which can be extended to **multiple-target CDR (MTCDR)**. Since utilizing information from multiple domains can improve performance, cross-domain recommendation has become an important scenario in recommender systems.

- **Multi-behavior Recommendation.** Users interact with recommender systems under multiple types of behaviors instead of only one type of behavior. For example, when a user clicks on the video, he/she may also perform behaviors such as collecting or commenting. In an e-commerce website, users often click, add into shopping carts, share, or collect a product before purchasing it, as shown in Figure 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 $\{y_1, y_2, \ldots, y_K\}$. For the ith behavior, if the user has an observed behavior, then $y_i = 1$, otherwise $y_i = 0$. The goal of the recommender system is to improve the prediction accuracy of a certain type of target behavior y_t.

  For multi-behavior recommendation, generally speaking, there are two main challenges. First, different behaviors have different influences on the target behavior. Some behaviors may be strong signals, and some may be weak signals [207, 208]. At the same time, this influence is different for each user. It is challenging to model the influence of these different behaviors on the target behavior accurately. Second, 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. 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 beyond-accuracy 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 that measures the dissimilarity of the recommended items for each user, since repeated similar items make users reluctant to explore the system. In other words,
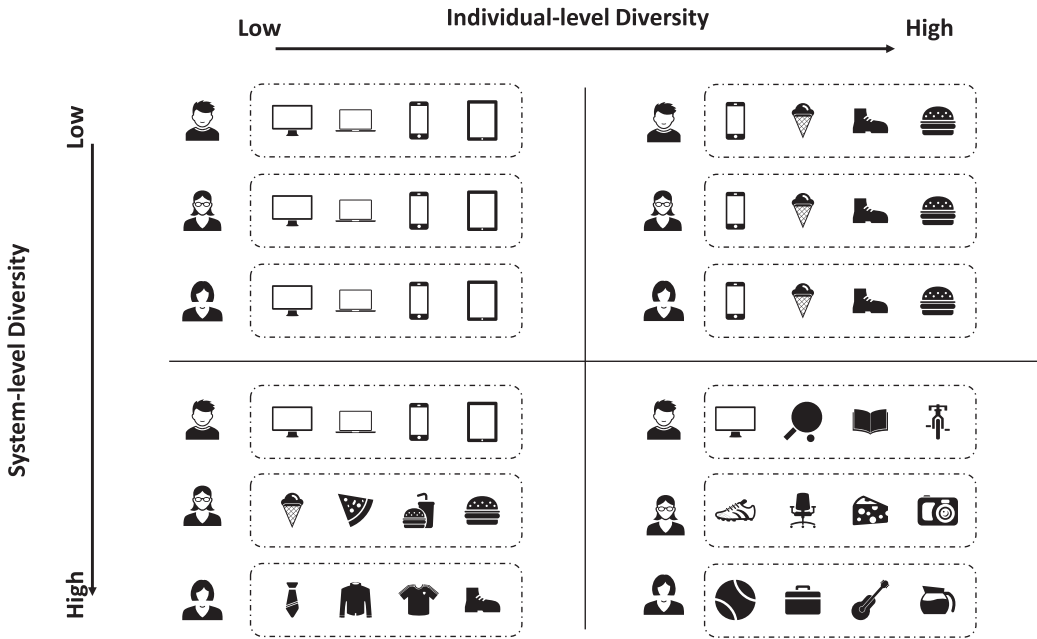
Fig. 9. Illustration of individual-level diversity and system-level diversity.

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 [242]. Regarding system-level diversity, it compares the recommendation results of different users and expects them to be dis-similar to 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. Figure 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 the two aspects.

- **Explainability.** As current recommender systems mostly adopt a deep learning paradigm, they can arouse urgent needs on the explainability of recommendation [236]. 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 [118, 184, 236]. Increasing the explainability of recommender systems can enhance users' perceived transparency [154], persuasiveness [166], and trustworthiness [86], and facilitate practitioners to debug and refine the system [236]. This survey mainly focuses on improving explainability with machine learning techniques. Specifically, past research adopts two different approaches [236]: 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 [237], Hidden Factor and Topic Model [126], and TriRank [64]. The others compromise slightly: They design post hoc separate models to explain the results generated by "black box" recommender systems, e.g., Explanation Mining [142]. 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 to 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, 41, 98, 128]. Specifically, according to the involved stakeholders, fairness in recommender systems can be divided into two categories [4, 98, 128]: user fairness, which attempts to ensure no algorithmic bias among specific users or demographic groups [9, 90, 96], and item fairness, which indicates fair exposures of different items, or no popularity bias among different items [2, 3, 98, 128]. 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, 125]. Specifically, researchers adopt two methods to enhance fairness: one is directly debiasing recommendation results in the training process [9, 254], while the other endeavors to rank items to alleviate the unfairness in a post-processing method [55, 153]. Indeed, increasing evidence shows that the utilization of graph data, e.g., user-user, could intensify concerns on fairness [36, 147, 195]. 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 exist 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 of 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 purchasing. Some works [122] propose to optimize the click-through rate and conversion rate at the same time. In addition, in e-commerce platform, products may have rich attributes, such as price [244], category [248], and so on, based on which heterogeneous graphs can be constructed [117]. Representative benchmark datasets for product recommendation include Amazon,[†] Tmall,[‡] and so on.

POI (Point-of-Interest) recommendation is also a popular application that aims to recommend new locations/point-of-interests for users' next visitation. In the 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 [46, 102, 189, 213]. Representative benchmark datasets for POI recommendation includes Yelp,[§] Gowalla,[¶] and so on.

[†]http://jmcauley.ucsd.edu/data/amazon.
[‡]https://tianchi.aliyun.com/dataset/dataDetail?dataId=649.
[§]https://www.yelp.com/dataset.
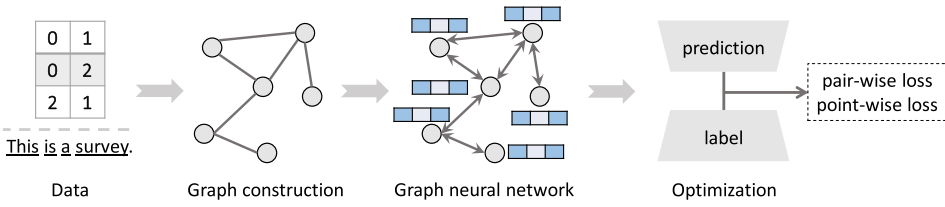[¶]https://snap.stanford.edu/data/loc-gowalla.html.

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.

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 [134]. Besides, users are always interested in up-to-date news and may refuse out-of-date ones. Thus, it is required to accurately filter news from the fast-changing candidate pools, which is quite challenging. As for news recommendation, MIND dataset [192] is the recently released representative benchmark dataset.

Movie recommendation is one of the earliest recommender systems. For example, Netflix's movie recommendation competition [7] motivated many pioneer recommendation researches [7, 84]. 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 [66, 149].

There are also other types of recommendation applications, such as video recommendation [11, 37, 110], music recommendation [87, 167, 186], job recommendation [139], food recommendation [131], and so on.

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, 47, 49, 83, 168, 229]. The rise of GNN mainly originates from the advancement of **convolutional neural network (CNN)** and **graph representation learning (GRL)** [205, 250]. 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 [144], learns node representations by using SkipGram [130] 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 Figure 10. Specifically, Sections 2.2.1, 2.2.2, and 2.2.3 elaborate on 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 node and edge.
- **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)** [220]. An emerging research direction in representation learning on graphs is **knowledge graph (KG)**, 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, and so on. Except for regular graphs, the hypergraph is also explored recently to handle more complex data (e.g., beyond pairwise relations and multiple modals) flexibly [47], 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 [152]. 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 the graph embeddings layer-by-layer.

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

- **GCN** [83]. 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 Reference [63]. Then the node

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 lth propagation layer
$\mathbf{A}$	The adjacency matrix of graph
$\mathbf{W}^l$	Learnable transformation matrix in the lth propagation layer
$\delta(\cdot)$	Nonlinear activation function
$\parallel$	Concatenation operation
$\odot$	Hadamard product

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 Reference [205]. $\mathbf{H}^l \in \mathbf{R}^{|V|\times D}$ is the embedding matrix of graph nodes in the lth 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** [62]. 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.

$$\mathbf{h}_{\mathcal{N}_i}^l = \text{AGGREGATE}_l\left(\{\mathbf{h}_j^l, \forall j \in \mathcal{N}_i\}\right),$$
$$\mathbf{h}_i^{l+1} = \delta\left(\mathbf{W}^l\left[\mathbf{h}_i^l\|\mathbf{h}_{\mathcal{N}_i}^l\right]\right), \tag{3}$$

where $\mathcal{N}_i$ denotes the sampled neighbors of the target node i. The function AGGREGATE has various options, such as MEAN, LSTM [68], and so on.

• **GAT** [168]. 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 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:

$$\mathbf{h}_i^{l+1} = \delta\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}^l\mathbf{h}_j^l\right),$$
$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}^l\mathbf{h}_i^l\|\mathbf{W}^l\mathbf{h}_j^l\right]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}^l\mathbf{h}_i^l\|\mathbf{W}^l\mathbf{h}_k^l\right]\right)\right)}, \tag{4}$$

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** [229]. 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
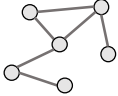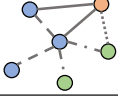
Model	Graph type of GNN	GNN category	Aggregate
GCN		Spectral	
GraphSAGE		Spatial	
GAT		Spatial	
HetGNN		Spatial	
HGNN		Spectral	

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.

the attention mechanism. Detailed formulas are omitted, since the implementation follows the above works.

- **HGNN** [47]. 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 are illustrated in Figure 11.

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, $L \leq 4$, due to the over-smoothing problem. Specifically, when the number of propagation layers is too large, the propagated embeddings will be too similar, losing distinguishing ability. Relevant studies focusing on developing deep and effective GNN models will be introduced in Section 5.1.1.

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 BPR [149] is as follows:

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

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, then 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, and 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 with by the user. Furthermore, those items that are interacted with 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 into 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 have 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 [78]. 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 Construction

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; reorganizing 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. However, it is also a challenging problem to determine whether to distinguish different types of nodes. For example, in the collaborative filtering task, user and item nodes can be modeled differently or considered the same kind of nodes. Another challenging point is handling concrete input such as numerical features like item prices, which are always continuous numbers. To represent these features in the graph, one possible solution is to discretize them into categorical ones, which can be then represented as nodes [244].
- **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 recommender system's data input can be considered 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. When constructing a graph, good designs of edges 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 huge 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 result in the poor utility of embedding propagation, since the propagation will be conducted on only a tiny fraction of nodes.

3.2 Network Design

The propagation layer makes GNN different from the traditional graph learning methods. 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, propagating item embeddings to a user node in the user-item interaction graph 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, and so on. 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 many users). Also, the propagation/aggregation layers can be stacked to help nodes access higher-hops neighbors. Too-shallow layers cannot model 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 [149] 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, their 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. Complex matrix operations are involved in each GCN layer, especially for the spectral GNN models such as GCN. 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 in 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 candidate selection, ranking for capturing user preferences, and re-ranking for other criteria beyond accuracy. The recommendation scenarios include social recommendation, sequential recommendation, cross-domain recommendation, and so on. 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. Tables 2, 3, and 4 show 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 candidate 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, and so on. For matching, besides the basic challenges of graph construction and network design, there is also the challenge of CF signal extraction, as shown in Table 5.

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, 160, 181, 183, 194]. 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. [181]

Table 2. A Summary of GNN-based Models in
Different Recommendation Stages in Top-tier Venues

Stage	Model	Venue	Year
Matching	GCMC [8]	KDD	2018
	PinSage [222]	KDD	2018
	NGCF [181]	SIGIR	2019
	LightGCN [65]	SIGIR	2020
	NIA-GCN [160]	SIGIR	2020
	DGCF [183]	SIGIR	2020
	IMP-GCN [104]	WWW	2021
	SGL [194]	SIGIR	2021
	HS-GCN [107]	TKDE	2022
	LGCN [228]	AAAI	2022
Ranking	Fi-GNN [99]	CIKM	2019
	PUP [244]	ICDE	2020
	A2-GCN [105]	TKDE	2020
	L_0-SIGN [158]	AAAI	2021
	DG-ENN [60]	KDD	2021
	TGIN [77]	WSDM	2022
Re-ranking	IRGPR [111]	CIKM	2020

proposed a spatial GNN in recommendation and obtain superior performance compared with conventional CF methods such as MF [85] or NCF [66]. Sun et al. [160] argued that simple aggregation mechanisms such as sum, mean, or max cannot model relational information among neighbors and proposed a neighbor interaction-aware convolution to address the issue. Wang et al. [183] 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. [194] leverage the stability of graph structure to incorporate a contrastive learning framework to assist representation learning. These GNN-based models can capture 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. However, the training complexity of GNN-based models was demonstrated [181, 183] as acceptable and comparable with non-graph models, especially when the transformation matrix is removed [65]. Besides, Reference [222] showed that the GNN-based model could be applied to web-scale recommender systems in real-world platforms efficiently and effectively, which combines random walk and GraphSAGE [62] for embedding learning on a large-scale item-item graph. Table 5 shows the commonality and difference among the GNN models in matching [104] proposed first to filter promising neighbors and then adopt embedding propagation.

We would like to briefly discuss the advantages and disadvantages of different GNN-based matching methods. These methods with sampling techniques such as Reference [222] can be used for large graphs, and they are more suitable for denser edges, such as item-item relations. Recent advances [65] remove non-linear operations to improve the performance further, but on some datasets, the performance of methods with non-linear operations [181] are still comparable. 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

Table 3. A Summary of GNN-based Models in Different
Recommendation Scenarios in Top-tier Venues

Scenario	Model	Venue	Year
Social	DiffNet [198]	SIGIR	2019
	GraphRec [43]	WWW	2019
	DANSER [201]	WWW	2019
	DGRec [157]	WSDM	2019
	HGP [82]	RecSys	2019
	DiffNet++ [197]	TKDE	2020
	MHCN [227]	WWW	2021
	SEPT [225]	KDD	2021
	GBGCN [230]	ICDE	2021
	KCGN [73]	AAAI	2021
	DiffNetLG [155]	SIGIR	2021
Sequential	ISSR [106]	AAAI	2020
	MA-GNN [119]	AAAI	2020
	STP-UDGAT [102]	CIKM	2020
	GPR [13]	CIKM	2020
	GES-SASRec [253]	TKDE	2021
	RetaGNN [69]	WWW	2021
	TGSRec [44]	CIKM	2021
	SGRec [97]	IJCAI	2021
	SURGE [15]	SIGIR	2021
Session	SR-GNN [203]	AAAI	2019
	GC-SAN [215]	IJCAI	2019
	TA-GNN [224]	SIGIR	2020
	MGNN-SPred [177]	WWW	2020
	LESSR [24]	KDD	2020
	MKM-SR [129]	SIGIR	2020
	GAG [146]	SIGIR	2020
	GCE-GNN [187]	SIGIR	2020
	SGNN-HN [137]	CIKM	2020
	DHCN [210]	AAAI	2021
	SHARE [173]	SDM	2021
	SERec [25]	WSDM	2021
	COTREC [209]	CIKM	2021
	DAT-MID [16]	SIGIR	2021
	TASRec [249]	SIGIR	2021
	G^3SR [40]	TNNLS	2022
	HG-GNN [138]	WSDM	2022
	CGL [136]	TOIS	2022
Bundle	BGCN [14]	SIGIR	2020
	HFGN [95]	SIGIR	2020
	BundleNet [38]	CIKM	2020
	DPR [246]	WWW	2021
	DPG [247]	TOIS	2022
	MIDGN [240]	AAAI	2022
Cross Domain	PPGN [238]	CIKM	2019
	BiTGCF [109]	CIKM	2020
	DAN [170]	CIKM	2020
	HeroGRAPH [35]	Recsys	2020
	DAGCN [58]	IJCAI	2021

Table 4. A Summary of GNN-based Models for Different
Recommendation Objectives in Top-tier Venues

Objective	Model	Venue	Year
Multi-behavior	MBGCN [78]	SIGIR	2020
	MGNN-SPred [177]	WWW	2020
	MGNN [235]	CIKM	2020
	LP-MRGNN [178]	TKDE	2021
	GNMR [206]	ICDE	2021
	MB-GMN [208]	SIGIR	2021
	KHGT [207]	AAAI	2021
	GHCF [17]	AAAI	2021
	DMBGN [212]	KDD	2021
Diversity	V2HT [94]	CIKM	2019
	BGCF [159]	KDD	2020
	DGCN [242]	WWW	2021
Explainability	RippleNet [171]	CIKM	2018
	EIUM [75]	MM	2019
	KPRN [185]	AAAI	2019
	RuleRec [121]	WWW	2019
	PGPR [211]	SIGIR	2019
	KGAT [180]	KDD	2019
	TMER [21]	WSDM	2021
Fairness	FairGo [195]	WWW	2021
	FairGNN [36]	WSDM	2021

Table 5. Details of GNN Models in Matching Stage

Model	Graph Construction	Network Design	Specific Design for Collaborative Filtering
GCMC [8]	user-item	GCN	weight sharing among relations
NGCF [181]	user-item	GCN	enhance propagated information
DGCF [183]	user-item	GAT	disentangled representations
LightGCN [65]	user-item	LightGCN	remove transformation and nonlinearity
SGL [194]	user-item	GCN	self-supervision on graphs
NIA-GCN [160]	user-item	NIA-GCN	neighbor interaction (NI)
PinSage [222]	item-item	GCN	neighbor sampling
IMP-GCN [104]	user-item	GCN	neighbor sampling
HS-GCN [107]	user-item	GCN	re-define distance metrics between nodes
LGCN [228]	user-item	low-pass GCN	trainable kernels for spectral graph convolution

usually first convert sparse features into one-hot encodings and then transform them into dense
embedding vectors. These feature embeddings are directly concatenated and fed into DNN [28, 57]
or specifically designed models [66, 148, 156] 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, as shown in Table 6. 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 to capture the desired feature interactions in
the encoder. On the 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.

Table 6. Details of GNN Models in Ranking Stage

Model	Graph Construction	Network Design	Feature Interaction
Fi-GNN [99]	fully connected	GAT+GRU	attention
PUP [244]	pre-defined	GCN	two-branch FM
L_0-SIGN [158]	learned	SIGN	graph classification
DG-ENN [60]	pre-defined	LightGCN	DNN
SHCF [91]	pre-defined	HGAT	inner product
GCM [193]	pre-defined	context GCN	FM
A2-GCN [105]	pre-defined	attentive GCN	inner product
TGIN [77]	pre-defined	triangle-based GCN	DNN

Li et al. [99] 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. [243, 244] 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. Liu [105] also approaches the recommendation problem when some item attributes are available and similarly builds item-attributes edges. Then the item embeddings can reflect both user behaviors and attribute information. Since not all feature interactions are useful, L_0-SIGN [158] automatically detects beneficial feature interactions and only reserves those edges, resulting in a learned graph that is further fed into a graph classification model to estimate the ranking score. In addition, Guo et al. [60] 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 [91] and GCM [193] 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 between the above GNN-based ranking models in terms of the designs of the encoder and predictor. Overall, these GNN-based ranking methods can be regarded as a combination of the traditional feature-interaction methods and GNN-based representation learning. As a result, the performance is still constrained by the feature-interaction component. A model that well couples the graph structure, such as high-order paths, into high-order feature extraction is desired.

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, and thus the major challenge is how to fuse multiple re-ranking goals. Liu et al. [111] 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 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. In short, the current works in GNN-based re-ranking only considered a few re-ranking goals, and others remain less explored.

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 [18, 161, 199, 200]. 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, a user's final behavior is decided by both the social impacts from friends and his/her preferences. One of the main challenges in social recommendation is constructing a social graph to capture the social influences of 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). Given that normal graph can only model the pairwise relations, normal graph-based methods [43, 61, 82, 116, 133, 155, 197, 198, 201, 217, 225, 226] stacked multiple GNN layers to capture multi-hop high-order social relations. However, stacked GNN layers may suffer from the over-smoothing [19] problem, which may lead to significant performance degradation. Hypergraph-based methods, such as MHCN [227], propose to model the high-order social relations with hyperedge [47], which can connect more than two nodes and model the high-order relations in a natural way. HOSR [113] recursively propagates embeddings on the social network to reflect the influence of high-order neighbors in the user representations. To further improve the recommendation performance, some works [6, 73, 157, 216, 230] introduce side information when constructing the graph. RecoGCN [216] unifies users, items, and selling agents into a heterogeneous graph to capture the complex relations in social E-commerce. GBGCN [230] constructs a graph for organizing user behaviors of two views in the group-buying recommendation, of which the *initiator view* contains those initiator-item interactions, and the *participant view* contains those participant-item interactions. DGRec [157] and TGRec [6] introduce temporal information of user behaviors into social recommendation. KCGN [73] 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., GCN and GAT. Some methods [6, 61, 73, 82, 155, 198, 217, 225, 227, 230] conduct GCN on social graph and treat the social influence of friends equally. RecoGCN [216] conducts meta-path-based GCN propagation on the constructed graph to capture both the social impact and user preference. HOSR [113] aggregates the information from neighbors with GCN to capture the high-order relations in the social graph. MHCN [227] performs propagation with GCN on constructed hypergraph to obtain the high-order social relations. Some methods with GAT mechanism [43, 116, 133, 157, 197, 201, 226], such as GraphRec [43] and DiffNet++ [197], 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

Table 7. Details of GNN Models for Social Recommendation

Model	Graph Construction	Network Design	Social Signal Extraction
DiffNet [198]	social graph	GCN	sum-pooling
GraphRec [43]	social graph + user-item graph	GAT	concatenation
DANSER [201]	social graph + user-item graph	GAT & GCN	–
DiffNet++ [197]	heterogeneous graph	GAT	multi-level attention network
MHCN [227]	multi-channel hypergraph + user-item graph	HyperGCN	sum-pooling
SEPT [225]	triangle-graphs + user-item graph	GCN	–
RecoGCN [216]	heterogeneous graph	Meta-path + GCN	concatenation
ESRF [226]	motif-induced graph	GAT	sum-pooling
GNN-SoR [61]	heterogeneous graph	GCN	concatenation
ASR [116]	heterogeneous graph	GAT	concatenation
GBGCN [230]	heterogeneous graph	GCN	–
DGRec [157]	social graph	GAT	–
SR-HGNN [217]	social graph + user-item graph	GCN	concatenation
KCGN [73]	social graph + item-item graph	GCN	concatenation
HGP [82]	group-user graph + user-item graph	GCN	attention mechanism
GAT-NSR [133]	social graph + user-item graph	GAT	MLP
HOSR [113]	social graph + user-item graph	GCN	attention mechanism
DiffNetLG [155]	heterogeneous graph	GCN	concatenation
TGRec [6]	heterogeneous graph	GCN	attention mechanism
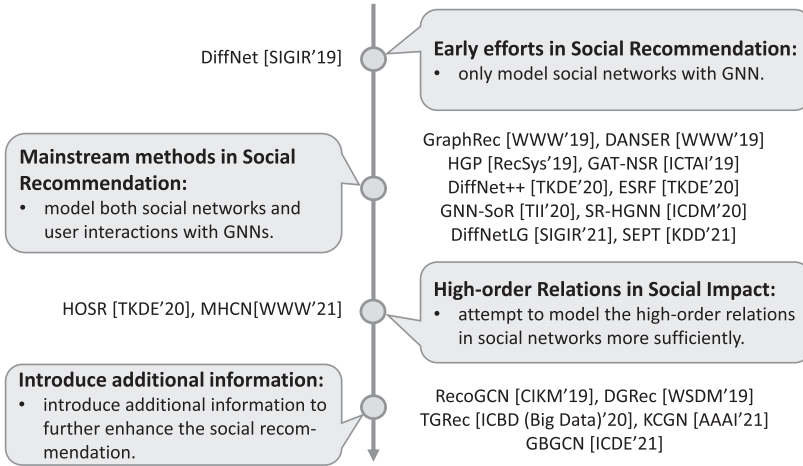
Fig. 12. Illustration of GNN models for social recommendation.

DiffNet [198], GraphRec [43], and MHCN [227], first separately learn user representations from social graph and user-item graph and then combine the representations with sum-pooling [198, 227], concatenation [43], MLP [133], or attention mechanism [6, 82, 113]. DiffNet++ [197], 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 Figure 12. Early efforts in social recommendation only model the social network with GNN, such as DiffNet [198]. Then, the methods [43, 61, 82, 116, 133, 155, 197, 198, 201, 217, 225, 226] that model both social network and user interactions with GNNs become the mainstream in

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

GNN-based social recommendation. Moreover, some studies, such as MHCN [227] and HOSR [113], attempt to enhance the recommendation by modeling the high-order relations in social networks more sufficiently. Also, there exist some works [6, 73, 157, 216, 230] that introduce additional information to further enhance the social recommendation. In short, the recent advances that can distinguish social relations with different strengths can always achieve better performance than early efforts roughly treating all social relations as the same.
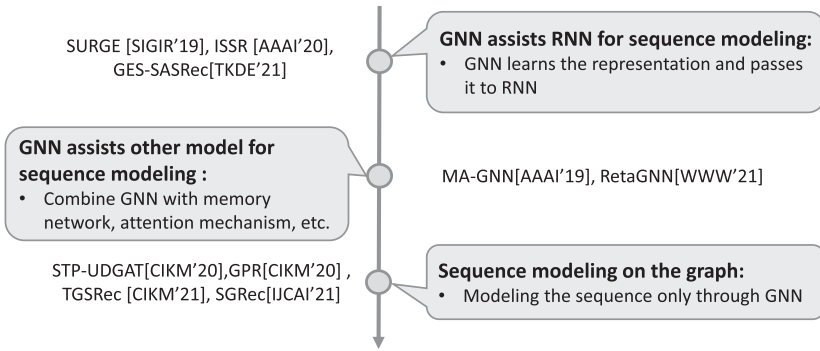
4.3.2 GNN in Sequential Recommendation. For sequential recommendation, to improve the recommendation performance, it is necessary to extract as much effective information as possible from the sequence and to learn the user's interest in the sequence, including short-term interest, long-term interest, dynamic interest, and so on, 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 [27] or recurrent neural networks [80]. Graph neural networks can be well leveraged for short-term, dynamic interest modeling or representation learning by converting the data to a graph. A general pattern for sequential modeling with GNN is shown in Figure 13.

SURGE [15] 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 flattening and finally used to predict the next item. Ma et al. [119] consider the short-term interest modeling in the sequence to build an item-item graph. For each item, this work 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 learning long-term user interests through other networks. The learned multiple representations are fused together and used for final recommendation.

Besides the challenges of graph construction and network design, the methods of GNN-based sequential recommendation should carefully design how to extract the sequential patterns in users' behaviors, as shown in Table 8. 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 ca not be accomplished by the Markov model or recurrent neural network. Wang et al. [169] 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. [106] 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,

Table 8. Details of GNN Models for Sequential Recommendation

| Model	Graph Construction	Network Design	Sequential Modeling
SURGE [15]	item-item graph	GAT	RNN
[169]	item-item graph	GCN	Attention
ISSR [106]	item-item and user-item graph	GCN	RNN
MA-GNN [119]	item-item graph	GCN	Memory network
DGSR [234]	user-item graph	GAT	RNN
GES-SASRec [253]	item-item graph	GCN	RNN
RetaGNN [69]	temporal heterogeneous graph	GAT	Self Attention
TGSRec [44]	temporal user-item graph	GAT	GAT
SGRec [97]	item-item graph	GAT	GAT
GME [213]	item-item and user-item graph	GAT	GAT
STP-UDGAT [102]	item-item and user-item graph	GAT	GAT
GPR [13]	item-item and user-item graph	GCN	GCN
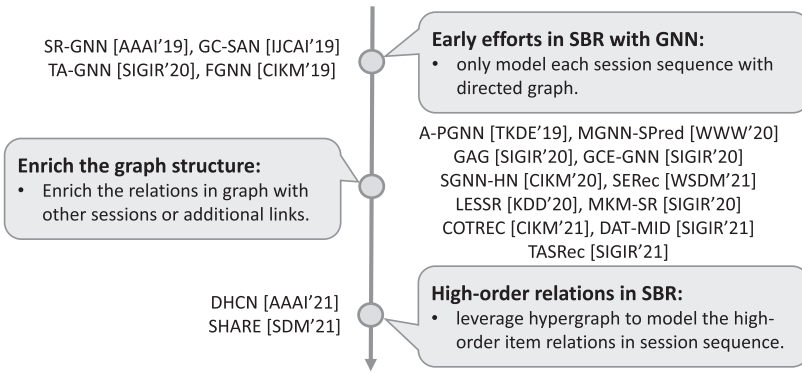
Fig. 14. Illustration of GNN models for session-based recommendation (SBR).

DGSR [234] and TGSRec [44] 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, along with the corresponding time attribute. Then perform convolution operations on the temporal graph to learn the representations of users and items. GES-SASRec [253] and SGRec [97] focus on the learning of item representations. For an item in a sequence, GES-SASRec [253] considers the next item of this item in other sequences, and SGRec [97] not only considers the next item but also considers the previous one. By aggregating the items before and after the target item in different sequences, the representation of the item is enhanced. GPR [13] and GME [213] construct 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 [69] considers the attributes of the items when constructing the graph, while STP-UDGAT [102] considers the geographic location, timestamp, and frequency in the POI recommendation. Table 8 summarizes the above works. Current networks of graph neural network-based recommendation are still highly relying on an additional sequence model such as RNN, Transformer, and so on, for sequential modeling. The ability of graphs in sequential modeling, such as directed edges, dynamic nodes, requires more research efforts.

4.3.3 GNN in Session-based Recommendation. In session-based recommendation, the session data may contain both user interests and noisy signals. Suppose a session for a certain user,

$iPhone \rightarrow iPad \rightarrow milk \rightarrow AirPods$. 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 the session-based recommendation are (1) how to model the item transition pattern in session data and (2) how to activate the 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 the user's actual interests.

- **Graph construction.** In session-based recommendation, most existing works [145, 187, 203, 215, 224] 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 [187, 210]. 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) adding the additional edges of the session graph. For the first strategy, A-PGNN [233], DGTN [245], and GAG [146] propose to enhance relations of the current session graph with related sessions, and GCE-GNN [187] leverages the global context by constructing another global graph to assist the transition patterns in the current session. DHCN [210] regards each session as a hyperedge and represents all sessions in a hypergraph to model the high-order item relations. SERec [25] enhances the global information for each session with a knowledge graph. CAGE [151] learns the representations of semantic-level entities by leveraging the open knowledge graph to improve the session-based news recommendation. MKM-SR [129] enhances the information in the given session by incorporating user micro-behaviors and item knowledge graph. COTREC [209] unifies all sessions into a global item graph from the item view and captures the relations among sessions by line graph from the session view. DAT-MID [16] follows GCE-GNN [187] to construct both session graph and global graph and then learns item embeddings from different domains. TASRec [249] 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 [137] constructs a star graph with a "star" node to gain extra knowledge in session data. SHARE [173] expands the hyperedge connections by sliding the contextual window on the session sequence. LESSR [24] proposes first to construct an edge-order preserving multigraph and then construct a shortcut graph for each session for enriching edge links. A^3SR [40] proposes to construct one global graph to represent all sessions and distinguish session-level prediction signals based on the later embedding propagation. HG-GNN [138] proposed to construct a heterogeneous graph in which the item transitions are captured by two types of item-item edge: in-type and out-type. CGL [136] also proposed to construct a global-level graph and a session-level graph, but it considers the learning on two graphs as two tasks, based on which multi-task learning is adopted.
- **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 [203] and its related works [25, 129, 137, 215, 224, 233] combine the gated recurrent units in the propagation (gated GNN) on the session graph. GAG [146], DCTN [245] conduct graph convolution on the constructed directed graph. DHCN [210] proposes to perform graph convolution on both hypergraph and line graph to obtain session representations from two different perspectives. Similar to

---

‖https://www.tmall.com.

Table 9. Details of GNN Models for Session-based Recommendation

Model	Graph Construction	Network Design	Enrich Graph Structure
SR-GNN [203]	directed graph	gated GNN	–
GC-SAN [215]	directed graph	gated GNN	–
TA-GNN [224]	directed graph	gated GNN	–
FGNN [145]	directed graph	GAT	–
A-PGNN [233]	directed graph	gated GNN	cross sessions
MGNN-SPred [177]	directed & multi-relational item graph	GraphSAGE	cross sessions
GAG [146]	directed graph	GCN	cross sessions
SGNN-HN [137]	star graph	gated GNN	additional edges
GCE-GNN [187]	directed graph + global graph	GAT	cross sessions
DGTN [245]	directed graph	GCN	cross sessions
DHCN [210]	hypergraph + line graph	HyperGCN	cross sessions
SHARE [173]	session hypergraph	HyperGAT	additional edges
SERec [25]	KG + directed graph	GAT + gated GNN	cross sessions
LESSR [24]	directed graph	GAT	additional edges
CAGE [151]	KG + article-level graph	GCN	cross sessions
MKM-SR [129]	KG + directed graph	gated GNN	cross sessions
COTREC [209]	item graph + line graph	GCN	cross sessions
DAT-MDI [16]	directed graph + global graph	GAT	cross sessions
TASRec [249]	dynamic graph	GCN	cross sessions
HG-GNN [138]	directed graph+global graph	gated GNN	cross sessions
CGL [136]	directed graph+global graph	gated GNN	cross sessions

DHCN [210], COTREC [209] performs GCN on item graph and line graph to obtain information from item and session views, respectively. CAGE [151] conducts GCN on article-level graph, and TASRec [249] performs graph convolution on the dynamic graph to capture the item relations. FGNN [145] conducts GAT on a directed session graph to assign different weights to different items. SHARE [173] performs GAT on session hypergraphs to capture the high-order contextual relations among items. GCE-GNN [187] and DAT-MID [16] perform GAT on both the session graph and global graph to capture the local and global information, respectively. MGNN-SPred [177] adopts GraphSAGE on a 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 Figure 14. Early efforts in SBR only model each session sequence with a directed graph, such as SR-GNN [203], GC-SAN [215], TA-GNN [224], and FGNN [145]. Then, some methods attempt to enrich the relation and information in the session graph with other sessions or additional links. The methods in References [16, 25, 129, 146, 151, 177, 187, 209, 210, 233, 245, 249] 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 [137], SHARE [173], and LESSR [24], 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 [210] and SHARE [173], attempt to enhance the recommendation by modeling the high-order relations in session data more sufficiently. As we can observe, the current works of GNN models for session-based recommendation have been developed from simple graphs to complex graphs, with even hyper-edges. Unlike sequential recommendation, session-based recommendation, in which users have repeated interactions with specific items, is more suitable for GNN-based models. Thus, the existing works of session-based recommendation can build more powerful and complex graph structures. A possible improvement of this area is to explore larger-scale datasets and further evaluate GNN's ability in these datasets.
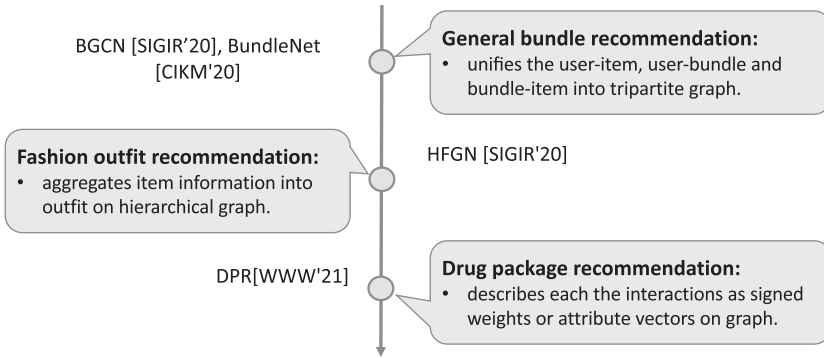
Fig. 15. Illustration of GNN models for bundle recommendation.

Table 10. Details of GNN Models for Bundle Recommendation

Model	Graph construction for affiliation	Refining bundle	Network design for High-order Relation
BGCN [14]	tripartite graph	meta-path propagation	multi-layer GCN
HFGN [95]	hierarchical graph	direct aggregation	multi-layer GCN
BundleNet [38]	tripartite graph	transformation parameters	multi-layer GCN
DPR [246]	tripartite graph	graph induction	multi-layer GCN
DPG [247]	tripartite graph	graph induction	multi-layer GCN
MIDGN [240]	tripartite graph	graph induction	multi-layer GCN & disentangling

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 existing works of GNN-based bundle recommendation are shown in Figure 15 by learning from user-item interaction and user-bundle interaction together, with parameter sharing or joint loss function [23, 141]. For the first time, Chang et al. [14] 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. [38] construct a similar tripartite graph with the transformation parameters to well extract bundle representations from included items' representations. Zheng et al. [246] 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. This method is further improved by Reference [247] by modeling the interaction data with the graph consisting of all bundles to obtain bundle embeddings and introducing a goal of new-bundle generation based on reinforcement learning. Li et al. [95] consider the problem of personalized outfit recommendation, which can also be regarded as a kind of bundle recommendation. The authors construct a hierarchical graph in which the users, items, and outfits are contained. GNN layers are deployed for obtaining the representations of users and outfits. The learning of the GNN model follows a multi-task manner. Zhao et al. [240] proposed a MIDGN model that combines the disentangle embedding space and multi-layer embedding propagation, obtaining finer-grained user preference towards items and bundles. The details of GNN models for bundle recommendation are shown in Table 10.

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. Benefitting from the powerful capabilities, the GNN-based recommendation model has gradually emerged in the cross-domain recommendation,
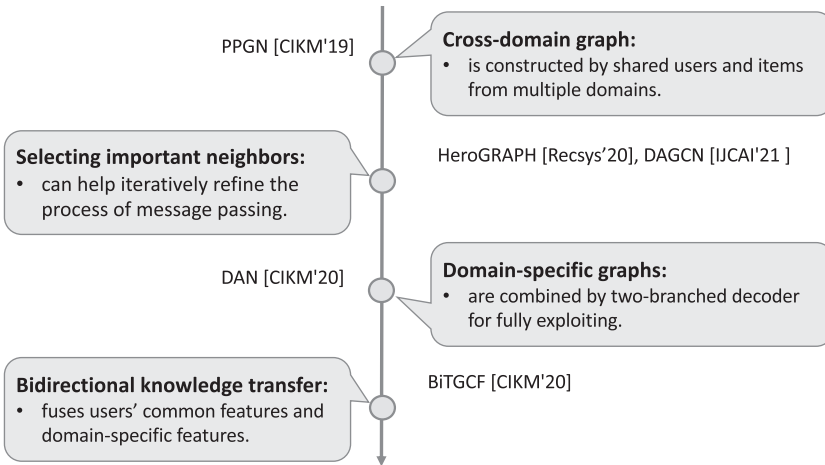
Fig. 16. Illustration of GNN models for cross-domain recommendation.

Table 11. Details of GNN Models of Cross-domain Recommendation

Model	Graph Construction	Network Design for Information Transferring
PPGN [238]	cross-domain graph	cross-domain propagation
BiTGCF [109]	domain-specific graphs	common user attributes
DAN [170]	domain-specific graphs	two-branched decoder
HeroGRAPH [35]	cross-domain graph	cross-domain propagation
DAGCN [58]	cross-domain graph	cross-domain propagation

as shown in Figure 16. The challenges include how to construct graph and design network architecture to transfer information across domains, as shown in Table 11. Zhao et al. [238] 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. [109] 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, and then it fuses common features with domain-specific features. Guo et al. [58] 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 adaptively choosing important neighbors during the embedding propagation. Wang et al. [170] 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. [35] proposes to construct a heterogeneous graph, where users and items in multiple domains 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 to distinguish important neighbors. To sum up, the cross-domain recommendation is a general definition, which covers various specific settings. For example, one setting may assume that the user can be overlapped, while another setting may be on the contrary.

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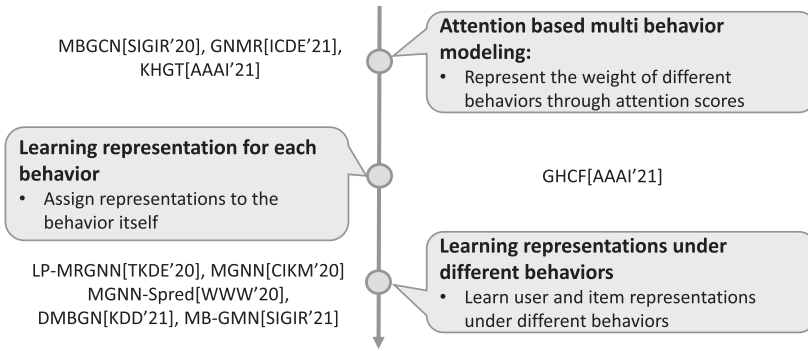
Fig. 17. Illustration of GNN models for multi-behavior recommendation.

recommender systems based on graph neural networks, based on standard 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 Figure 17.

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. [235] construct all user behaviors in one graph and perform graph convolution operations. Wang [177, 178] 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. [17] proposed an intuitive method that assigns a representation to users, items, and behaviors. In the propagation process, 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. [206] also redesign the aggregation mechanism on the graph convolutional network to explicitly model the impact of different types of behavior. Jin et al. [78] assign different learnable weights to different edges to model the importance of the behaviors. In addition, to capture complex multi-behavior relationships, some works rely on knowledge. For example, Xia et al. [207] learn 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. [208] use 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, then the representation of the item can be enhanced. To get better item representation, Yu et al. [223] not only connect related items in the graph but also construct 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 meta-path or second-order neighbors,

Table 12. Details of GNN Models for Multi-behavior Recommendation

Model	Graph Construction	Network Design	Multi-behavior Modeling
MB-GMN [208]	user-item graph	GCN / GAT	Behavior representation
KHGT [207]	user-item/item-item graph	GAT	Weight
MBGCN [78]	user-item graph	GAT	Weight
MGNN-SPred [177]	item-item graph	GCN	Weight
MGNN [235]	user-item graph	GCN	Node representation
LP-MRGNN [178]	item-item graph	GCN	Weight
GNNH [223]	item-item/category-category graph	GCN	Item representation
GNMR [206]	user-item graph	GAT	Weight
DMBGN [212]	item-item graph	GCN	Graph representation
GHCF [17]	user-item graph	GCN	Edge embedding

where similar items are aggregated to enhance their representation [78, 207]. These works' details are presented in Table 12.

4.4 GNN for Different Recommendation Objectives

4.4.1 GNN for Diversity. For 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 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, 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. [159] propose a model called **Bayesian Graph Collaborative Filtering (BGCF)**, which constructs augmented graphs with node copying [135] from high-order neighbors, such that items of diverse topics with high similarity can be directly connected to user nodes. Zheng et al. [242] 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. Figure 19 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 of item categories, which further improves recommendation diversity. The comparisons are illustrated in Figure 18.

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 than those popular items. To address the weak signals of long-tail items, Li et al. [94] 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 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.
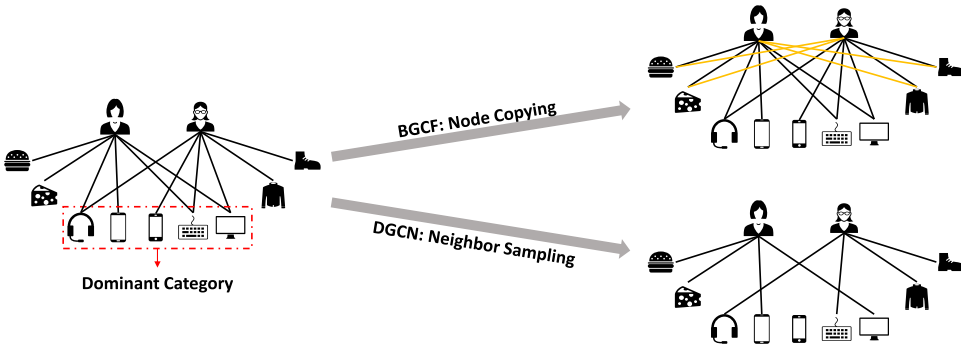
Fig. 18. Illustration of BGCF [159] and DGCN [242] on achieving individual-level diversity.
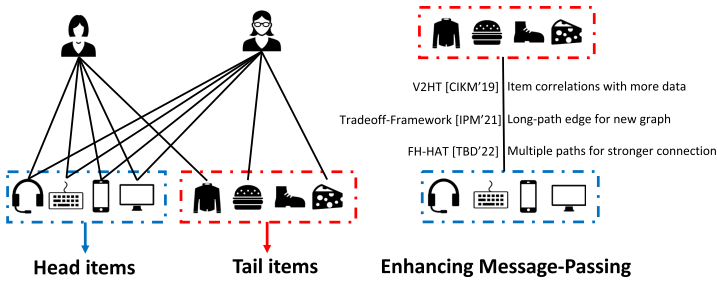

Fig. 19. Illustration of V2HT [94], FH-HAT [214], and Tradeoff-Framework [76] on achieving system-level diversity.

Table 13. Details of GNN Models for Diversified Recommendation (How They Address Challenges)

Model	Weak Preference Signals	Accuracy-diversity Balance
BGCF [159]	Node Copying	Re-rank
DGCN [242]	Neighbor Sampling	Adversarial training
V2HT [94]	Utilize item correlations	-
FH-HAT [214]	Heterogeneous graph	Diversity loss
Tradeoff-Framework [76]	NN & FN graph	Joint training

In addition, a few studies [76, 214] utilize GNN to improve both individual-level and system-level diversity. Specifically, Xie et al. [214] 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 accuracy and diversity. Isufi et al. [76] 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 achieve a tradeoff between accuracy and diversity. Table 13 shows the differences among the above approaches.

4.4.2 GNN for Explainability. With the proliferation of GNN, researchers also make endeavors to improve the explainability of recommender systems with GNN's power of modeling logical 1000 relations, as illustrated in Figure 20. He et al. [64] 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
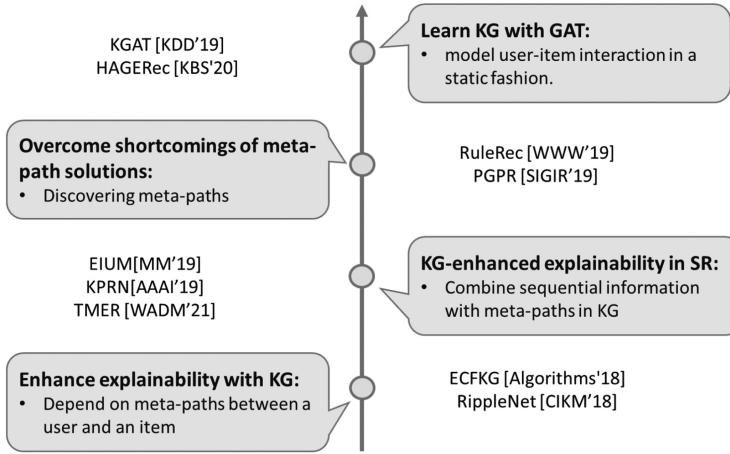
Fig. 20. Illustration of GNN models for explainable recommendation.

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 the form of knowledge graph, to enhance the explainability [236]. 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 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. [171] proposed an end-to-end framework, RippleNet, that combines the two knowledge graph-aware recommendation methods together. Here, the knowledge graph contains the related knowledge of the recommended items, such as 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 [75, 185]. Huang et al. [75] 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. [185] also leverage 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. [21] 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 [121, 211]. First, since pre-defined meta-paths require extensive domain knowledge, Ma et al. [121] 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. [211] replace the enumeration method with the reinforcement reasoning approach to identify proper meta-paths for scalability.

Besides meta-path-based solutions, Wang et al. [180] propose a new method named **Knowledge Graph Attention Network (KGAT)**, where the attention mechanism can offer explainability to

Table 14. Details of GNN Models for Explainable Recommendation

Model	Network Design	Source of Explainability
ECFKG [5]	Meta-path based GE	Meta-path over knowledge graph
RippleNet [171]	Meta-path based GraphSage	Meta-path over knowledge graph
EIUM [75]	Meta-path based GE	Meta-path over knowledge graph
KPRN [185]	Meta-path based GE	Meta-path over knowledge graph
TMER [21]	Meta-path based GE	Meta-path over knowledge graph and temporal dependency
RuleRec [121]	Meta-path based GE	Discovered meta-path over knowledge graph
PGPR [211]	Meta-path based GE	Discovered meta-path over knowledge graph
KGAT [180]	GAT	Attention mechanism
HAGERec [219]	GAT	Attention mechanism

some extent. Yang et al. [219] 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. Explainability in recommendation includes designing explainable model and finding explanations for those models without explainability (such as a black-box model). That is, the explainable model can naturally generate explainable recommendation results. As for GNN-based recommendation models for explainability, they can utilize paths on heterogenous graphs or knowledge graphs to generate recommendation. Since the model architecture is explicit and explainable, they are classified into the first category of explainability in recommendation. We compare the technical details in Table 14. The existing works of GNN-based recommendation for explainability mostly extract the relations between attributes or behaviors with recommendation results, which ignores the causality, which may lead to wrong explanations. Designing suitable causal inference methods into graph neural network-based models is important and promising.

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 [36, 147, 195]. 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 [36].

To curb the fairness issue in recommendation, some researchers propose to learn fair graph embeddings [10, 147, 195]. Rahman et al. [147] extend the well-known graph embedding method, node2vec [56], to a more fair version, Fairwalk, which can generate a 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] propose 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 vary, fairness constraints are flexible according to the task. However, similar to Fairwalk [147], all nodes' sensitive attributes are required. Wu et al. [195] 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. [36] 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 of sensitive data. Moreover, they develop GNN-based sensitive data estimators to overcome the issue of missing sensitive data in the real world. We compare the technical details in Table 15.

Table 15. Details of GNN Models for Fair Recommendation

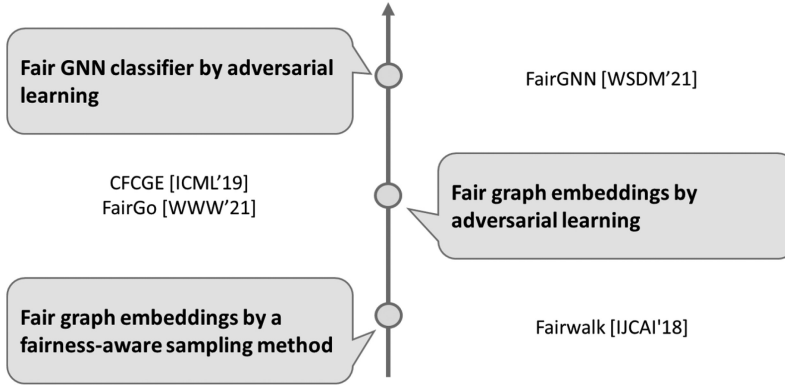| Model	Network Design	Achieving Fairness
Fairwalk [147]	Node2vec (GE)	learning fair graph embeddings by sampling
CFCGE [10]	Invariant GE	learning fair graph embeddings by adversarial learning
FairGo [195]	HetGNN(GNN)	learning fair graph embeddings by adversarial learning
FairGNN [36]	GraphSage(GNN)	learning fair GNN classifiers by adversarial learning

Fig. 21. Illustration of GNN models for fair recommendation.

In short, existing GNN-based recommendation models for achieving fairness, as shown in Figure 21, always lead to performance drop to some extent, requiring solutions for addressing the tradeoff dilemma.

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. [100] 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. [112] 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 [13] 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. [231] propose to combine social networks and user-item interactions together, which deploys embedding aggregation on both social-connected users and visited POIs. As for news recommendation, Hu et al. [71] propose to introduce the preference disentanglement [179] into the user-news embedding propagation.

There are many other 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 More Complex GNNs

5.1.1 Deeper GNN. 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 [20, 88, 150, 251]. Despite these advancements, there is no

universal solution for constructing very deep GNN like CNN, and relevant works propose different strategies. Lai et al. [88] develop 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 propagation layers to boost model performance. Rong et al. [150] alleviate the over-smoothing problem by randomly removing graph edges, which acts as a message-passing reducer. Claudio et al. [51] consider 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 and go deeper. Li et al. [92] 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 a fundamental problem in developing very deep GNNs, like groundbreaking works in the area of CNN [74, 164]. At the same time, the computation and time complexity must also be acceptable.

5.1.2 Dynamic GNN. Existing GNN-based recommendation models are almost based on the static graph, as mentioned above, while there are 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, and so on, which poses challenges to static graph neural networks. Recently, dynamic graph neural networks [93, 123] have attracted attention, which deploys embedding propagation operations 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.

5.1.3 Efficiency and Scalability in Larger GNN. 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. A recent work [239] even proposes a GNN-based solution for both the searching and recommendation engines, which results in higher requirements of computational efficiency. Hamilton et al. propose GraphSAGE [62], 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 [22, 29], which reduces the computation complexity of GNN and improves the scalability. Ying et al. [222] successfully apply GraphSAGE to web-scale recommender systems, which can efficiently compute embeddings for billions of items. Some other works [45] design specific query-recall acceleration strategies for an efficient recommendation. In addition, a few open-source tools have been released that can accelerate the research and development of GNN-based recommendation, such as PyG [48], DGL [174], and AliGraph [252]. Recently, a graph learning system with both effectiveness and scalability, PlatoGL [103], has been proposed. It provides a practical solution to combine large-scale GNN with real-time recommendation, a fundamental problem for today's industrial recommendations. We refer to another survey [1] for details on computing and accelerating GNN.

Despite these existing approaches, achieving 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.

5.2 Advanced Machine Learning Methods

5.2.1 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. [227] and Wu et al. [194] 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 it is a promising future direction to leverage extra self-supervised tasks to learn meaningful and robust representations for GNN-based recommender systems.

5.2.2 AutoML-enhanced GNN. Recommendation scenarios are diverse and vastly different from each other; thus, there exists no silver bullet GNN model that can generalize across all scenarios. Recently, **AutoML (Automated Machine Learning)** [221] has been proposed, which can automatically design appropriate models for specific tasks. For GNN-based recommendation, the search space is quite large, including multiple options for the neighbor sampler, aggregator, interaction function, and so on. Consequently, AutoML can, to a great extent, reduce human effort in discovering advanced model structures. A few works [54, 72] have been proposed that search to combine GNN layers and aggregate neighbors. Therefore, designing AutoML algorithms to search for GNN-based recommender systems [188] is a promising future direction.

5.3 Others

5.3.1 About the Taxonomy. In this survey, we have presented a taxonomy for GNN-based recommendation, in which we discuss the existing works from four aspects: stage, scenario, objective, and application. Here, we discuss the important future directions following the presented taxonomy.

- **Stage.** The traditional recommendation divides the whole recommendation process into several stages mainly due to the too-large size of the original candidate pool (million or even billion). That is, different stages of recommender engines have different requirements for models. Currently, with the fast growth of computing power, the boundaries between different stages get more and more blurred, which can be understood in two folds. First, the model in the ranking stage starts to take a huge-size item pool and multiple user behaviors into consideration. Second, the multiple objectives in the re-ranking stage, such as diversity, can also be achieved by the model of the recall stage or ranking stage, making the re-ranking stage not so necessary.
- **Scenario.** With the fast development of online information services, recommendation scenarios are also changing rapidly. For example, the user can interact with the recommender systems more actively by chatting with the robot, defined as a new scenario of conversational recommendation, which we will discuss in detail later. Recently, some works [239] consider the search and recommendation simultaneously, two core concepts of information retrieval, which try to design a unified graph neural network model to provide two kinds of services.
- **Objectives.** Existing works of GNN-based recommendation models have spent much effort in achieving various objectives, including accuracy, diversity, explainability, fairness, and so on. However, they are still suffering from the tradeoff dilemma, in which only one goal can be optimized simultaneously. Recently, some works of deep learning-based recommendation, such as MMOE [120], PLE [165], and so on, studied how to deploy multiple separate models, each of which is designed towards one objective, and then optimize them together. We

believe that these techniques in multi-objective learning can also be well used in addressing the tradeoff in GNN-based recommendation models.

- **Applications.** On the one hand, a powerful recommender system helps existing non-personalized applications attract more users and promote platform profits. On the other hand, relying on its benefit of both user-side and platform-side, the recommender system can motivate some new forms of applications, such as micro-video recommendation in TikTok. In these applications where content has abundant features, deploying cross-modal graph neural network models in both content understanding and user behavior modeling is a promising research direction.

5.3.2 KG-enhanced Recommendation with GNN. Recent research has demonstrated the power of KG in recommendation by enriching the user-item bipartite graph with knowledge [53]. Specifically, the utilization of knowledge graph with GNN significantly addresses some practical issues in recommender systems [53], for example, cold start problem [42, 232] and dynamicity [163]. Moreover, knowledge graph also offers a novel solution to some scenarios in recommendations, e.g., sequential recommendation [75, 185], and objectives, e.g., explainability [140, 180, 219]. There are already many works, including KGCN [172], KGAT [180], KGIN [182], and so on. Currently, the majority of research first uses GNN to learn embeddings of knowledge graphs and then incorporate these embeddings into the recommendation model so an end-to-end model can be trained [169]. Therefore, we point out that the utilization of KG in recommendation with GNN can be further enhanced from the perspectives of data, scenario, and model. Specifically, the incorporated KG mainly records the rich item-item relations, e.g., Movie A belongs to Category B [53], but user-user knowledge is lacking in formal and plausible definition and thus substantially overlooked. Future work can consider creating user-centric KG based on abundant knowledge in sociology. Moreover, considering its success in scenarios, such as sequential recommendation [75, 185], 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 of 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.

5.3.3 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 [89, 115, 162, 204] proposed a new paradigm in which users interact with the information system through 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 for future work, the advances in representation learning with graph neural networks can be combined with preference learning in the conversational recommendation.

6 CONCLUSION

There is a rapid development of graph neural network models in the research field of recommender systems. This article provides an extensive survey systematically presenting the challenges, methods, and future directions in this area. Not only the history of development but 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. 2021. Computing graph neural networks: A survey from algorithms to accelerators. *Comput. Surv.* 54, 9 (2021), 1–38.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the 11th 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. In *Proceedings of the RecSys Workshop on Recommendation in Multistakeholder Environments (RMSE)*. 1–7.

[4] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2020. The connection between popularity bias, calibration, and fairness in recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 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 *Proceedings of the IEEE International Conference on Big Data*. 898–903.

[7] Robert M. Bell and Yehuda Koren. 2007. Lessons from the Netflix prize challenge. *ACM SIGKDD Explor. Newslett.* 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 KDD Deep Learning Day*. 1–7.

[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*. 2212–2220.

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

[11] Desheng Cai, Shengsheng Qian, Quan Fang, Jun Hu, Wenkui Ding, and Changsheng Xu. 2022. Heterogeneous graph contrastive learning network for personalized micro-video recommendation. *IEEE Trans. Multim.* (2022), 1–1.

[12] 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 *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 585–594.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[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] 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.

[21] 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*. 1056–1064.

[22] Jie Chen, Tengfei Ma, and Cao Xiao. 2018. FastGCN: Fast learning with graph convolutional networks via importance sampling. In *Proceedings of the International Conference on Learning Representations*. 1–15.

[23] 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 *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2095–2101.

[24] 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.

[25] 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.

[26] Xu Chen, Yongfeng Zhang, and Ji-Rong Wen. 2022. Measuring why in recommender systems: A comprehensive survey on the evaluation of explainable recommendation. *arXiv preprint arXiv:2202.06466* (2022).

[27] Chen Cheng, Haiqin Yang, Michael R. Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 2605–2611.

[28] 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.

[29] 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.

[30] Janneth Chicaiza and Priscila Valdiviezo-Diaz. 2021. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information* 12, 6 (2021), 232.

[31] Minjin Choi, Jinhong Kim, Joonseok Lee, Hyunjung Shim, and Jongwuk Lee. 2022. S-walk: Accurate and scalable session-based recommendation with random walks. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 150–160.

[32] Yunfei Chu, Jiangchao Yao, Chang Zhou, and Hongxia Yang. 2022. Graph neural networks in modern recommender systems. In *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer, 423–445.

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

[34] 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.

[35] Qiang Cui, Tao Wei, Yafeng Zhang, and Qing Zhang. 2020. HeroGRAPH: A heterogeneous graph framework for multi-target cross-domain recommendation. In *Proceedings of ORSUM@ RecSys: 3rd Workshop on Online Recommender Systems and User Modeling*.

[36] Enyan Dai and Suhang Wang. 2021. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 680–688.

[37] 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 4th ACM Conference on Recommender Systems*. 293–296.

[38] 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.

[39] Yue Deng. 2021. Recommender systems based on graph embedding techniques: A comprehensive review. *arXiv preprint arXiv:2109.09587* (2021).

[40] Zhi-Hong Deng, Chang-Dong Wang, Ling Huang, Jian-Huang Lai, and Philip S. Yu. 2022. G^{3}SR: Global graph guided session-based recommendation. *IEEE Trans. Neural Netw. Learn. Syst.* (2022), 1–14.

[41] Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. 2022. Online certification of preference-based fairness for personalized recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 6532–6540.

[42] 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*. 2478–2486.

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

[44] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S. Yu. 2021. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 433–442.

[45] Chao Feng, Defu Lian, Xiting Wang, Zheng Liu, Xing Xie, and Enhong Chen. 2023. Reinforcement routing on proximity graph for efficient recommendation. *ACM Trans. Inf. Syst.* 41, 1 (2023), 1046–8188.

[46] 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 *Proceedings of the 24th International Joint Conference on Artificial Intelligence.*

[47] 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.

[48] Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with PyTorch geometric. In *Proceedings of the Workshop of International Conference on Learning Representations.* 1–9.

[49] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. 2017. Protein interface prediction using graph convolutional networks. *Adv. Neural Inf. Process. Syst.* 30 (2017), 1–10.

[50] 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.

[51] 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.

[52] 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 *Proceedings of the World Wide Web Conference.* 491–502.

[53] 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).

[54] Yang Gao, Hong Yang, Peng Zhang, Chuan Zhou, and Yue Hu. 2021. Graph neural architecture search. In *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence.* 1403–1409.

[55] 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.* 2221–2231.

[56] 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.

[57] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence.* 1725–1731.

[58] 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. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence.* 2483–2489.

[59] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2022. A survey on knowledge graph-based recommender systems. *IEEE Trans. Knowl. Data Eng.* 34, 8 (2022), 3549–3568.

[60] 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 CTR prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 496–504.

[61] Zhiwei Guo and Heng Wang. 2020. A deep graph neural network-based mechanism for social recommendations. *IEEE Trans. Industr. Inform.* 17, 4 (2020), 2776–2783.

[62] 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.

[63] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. 2011. Wavelets on graphs via spectral graph theory. *Appl. Computat. Harmon. Anal.* 30, 2 (2011), 129–150.

[64] 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.

[65] 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.

[66] 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.

[67] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations.* 1–10.

[68] Sepp Hochreiter and Júrgen Schmidhuber. 1997. Long short-term memory. *Neural Computat.* 9, 8 (1997), 1735–1780.

[69] Cheng Hsu and Cheng-Te Li. 2021. RetaGNN: Relational temporal attentive graph neural networks for holistic sequential recommendation. In *Proceedings of the Web Conference.* 2968–2979.

[70] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative cross networks for cross-domain recommen-
      dation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 667–676.
[71] 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.
[72] Zhao Huan, Yao Quanming, and Tu Weiwei. 2021. Search to aggregate neighborhood for graph neural network. In
      Proceedings of the IEEE 37th International Conference on Data Engineering. 552–563.
[73] 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 *Proceedings of the
      AAAI Conference on Artificial Intelligence*.
[74] 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.
[75] 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 Inter-
      national Conference on Multimedia*. 548–556.
[76] Elvin Isufi, Matteo Pocchiari, and Alan Hanjalic. 2021. Accuracy-diversity trade-off in recommender systems via
      graph convolutions. *Inf. Process. Manag.* 58, 2 (2021), 102459.
[77] Wensen Jiang, Yizhu Jiao, Qingqin Wang, Chuanming Liang, Lijie Guo, Yao Zhang, Zhijun Sun, Yun Xiong, and
      Yangyong Zhu. 2022. Triangle graph interest network for click-through rate prediction. In *Proceedings of the 15th
      ACM International Conference on Web Search and Data Mining*. 401–409.
[78] 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.
[79] 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.
[80] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the IEEE
      International Conference on Data Mining*. IEEE, 197–206.
[81] Wang-Cheng Kang and Julian McAuley. 2019. Candidate generation with binary codes for large-scale Top-N rec-
      ommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.
      1523–1532.
[82] 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 so-
      cial recommendation. *arXiv preprint arXiv:1908.02569* (2019).
[83] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Pro-
      ceedings of the International Conference on Learning Representations*. 1–14.
[84] Yehuda Koren. 2009. The BellKor solution to the Netflix grand prize. *Netflix Prize Docum.* 81, 2009 (2009), 1–10.
[85] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems.
      Computer 42, 8 (2009), 30–37.
[86] 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 CHI Conference on
      Human Factors in Computing Systems*. 1–12.
[87] Valerio La Gatta, Vincenzo Moscato, Mirko Pennone, Marco Postiglione, and Giancarlo Sperlí. 2022. Music recom-
      mendation via hypergraph embedding. *IEEE Trans. Neural Netw. Learn. Syst.* (2022), 1–13.
[88] 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.
[89] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. In-
      teractive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th ACM SIGKDD
      International Conference on Knowledge Discovery & Data Mining*. 2073–2083.
[90] Jurek Leonhardt, Avishek Anand, and Megha Khosla. 2018. User fairness in recommender systems. In *Proceedings of
      the Web Conference*. 101–102.
[91] Chen Li, Linmei Hu, Chuan Shi, Guojie Song, and Yuanfu Lu. 2021. Sequence-aware heterogeneous graph neural
      collaborative filtering. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, 64–72.
[92] 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.

[93] 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.

[94] 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.

[95] 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.

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

[97] Yang Li, Tong Chen, Yadan Luo, Hongzhi Yin, and Zi Huang. 2021. Discovering collaborative signals for next POI recommendation with iterative Seq2Graph augmentation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. 1491–1497.

[98] Yunqi Li, Yingqiang Ge, and Yongfeng Zhang. 2021. Tutorial on fairness of machine learning in recommender systems. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[99] 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.

[100] 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 *Proceedings of the IEEE 36th International Conference on Data Engineering*. 1677–1688.

[101] 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.

[102] 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.

[103] Dandan Lin, Shijie Sun, Jingtao Ding, Xuehan Ke, Hao Gu, Xing Huang, Chonggang Song, Xuri Zhang, Lingling Yi, Jie Wen, and Chuan Chen. 2022. PlatoGL: Effective and scalable deep graph learning system for graph-enhanced real-time recommendation. In *Proceedings of the Conference on Information and Knowledge Management*.

[104] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-aware message-passing gcn for recommendation. In *Proceedings of the Web Conference*. 1296–1305.

[105] Fan Liu, Zhiyong Cheng, Lei Zhu, Chenghao Liu, and Liqiang Nie. 2022. A2-GCN: An attribute-aware attentive GCN model for recommendation. *IEEE Trans. Knowl. Data Eng.* 34, 9 (2022), 4077–4088.

[106] Feng Liu, Weiwen Liu, Xutao Li, and Yunming Ye. 2020. Inter-sequence enhanced framework for personalized sequential recommendation. *arXiv preprint arXiv:2004.12118* (2020).

[107] Han Liu, Yinwei Wei, Jianhua Yin, and Liqiang Nie. 2022. HS-GCN: Hamming spatial graph convolutional networks for recommendation. *IEEE Trans. Knowl. Data Eng.* (2022), 1–1.

[108] Jiawei Liu, Chuan Shi, Cheng Yang, Zhiyuan Lu, and Philip S. Yu. 2022. A survey on heterogeneous information network based recommender systems: Concepts, methods, applications and resources. *AI Open* 3 (2022), 40–57.

[109] 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.

[110] 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.

[111] Weiwen Liu, Qing Liu, Ruiming Tang, Junyang Chen, Xiuqiang He, and Pheng Ann Heng. 2020. Personalized reranking with item relationships for e-commerce. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 925–934.

[112] Weiwen Liu, Yin Zhang, Jianling Wang, Yun He, James Caverlee, Patrick P. K. Chan, Daniel S. Yeung, and Pheng-Ann Heng. 2022. Item relationship graph neural networks for e-commerce. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 9 (2022), 4785–4799.

[113] Yang Liu, Chen Liang, Xiangnan He, Jiaying Peng, Zibin Zheng, and Jie Tang. 2022. Modelling high-order social relations for item recommendation. *IEEE Trans. Knowl. Data Eng.* 34, 9 (2022), 4385–4397.

[114] Yidan Liu, Min Xie, and Laks V. S. Lakshmanan. 2014. Recommending user generated item lists. In *Proceedings of the 8th ACM Conference on Recommender Systems*. 185–192.

[115] Zeming Liu, Ding Zhou, Hao Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, Ting Liu, and Hui Xiong. 2022. Graph-grounded goal planning for conversational recommendation. *IEEE Trans. Knowl. Data Eng.* (2022), 1–1.

[116] Dongsheng Luo, Yuchen Bian, Xiang Zhang, and Jun Huan. 2020. Attentive social recommendation: Towards user and item diversities. *arXiv preprint arXiv:2011.04797* (2020).

[117] 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 ACM SIGMOD International Conference on Management of Data*. 313–327.

[118] Ziyu Lyu, Yue Wu, Junjie Lai, Min Yang, Chengming Li, and Wei Zhou. 2022. Knowledge enhanced graph neural networks for explainable recommendation. *IEEE Trans. Knowl. Data Eng.* (2022), 1–1.

[119] 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.

[120] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1930–1939.

[121] 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 *Proceedings of the World Wide Web Conference*. 1210–1221.

[122] 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 *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[123] 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.

[124] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: An embedding and mapping approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2464–2470.

[125] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Fair-Match: A graph-based approach for improving aggregate diversity in recommender systems. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. 154–162.

[126] 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.

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

[128] 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*. 2243–2251.

[129] 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.

[130] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*. 1–12.

[131] Weiqing Min, Shuqiang Jiang, and Ramesh Jain. 2019. Food recommendation: Framework, existing solutions, and challenges. *IEEE Trans. Multim.* 22, 10 (2019), 2659–2671.

[132] Nima Mirbakhsh and Charles X. Ling. 2015. Improving Top-N recommendation for cold-start users via cross-domain information. *ACM Trans. Knowl. Discov. Data* 9, 4 (2015), 1–19.

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

[134] 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.

[135] Soumyasundar Pal, Florence Regol, and Mark Coates. 2019. Bayesian graph convolutional neural networks using node copying. In *Proceedings of the ICML Workshop on Learning and Reasoning with Graph-structured Data*.

[136] Zhiqiang Pan, Fei Cai, Wanyu Chen, Chonghao Chen, and Honghui Chen. 2022. Collaborative graph learning for session-based recommendation. *ACM Trans. Inf. Syst.* 40, 4 (2022), 1–26.

[137] 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.

[138] Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. 2022. Heterogeneous global graph neural networks for personalized session-based recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 775–783.

[139] Ioannis Paparrizos, B. Barla Cambazoglu, and Aristides Gionis. 2011. Machine learned job recommendation. In *Proceedings of the 5th ACM Conference on Recommender Systems*. 325–328.

[140] Sung-Jun Park, Dong-Kyu Chae, Hong-Kyun Bae, Sumin Park, and Sang-Wook Kim. 2022. Reinforcement learning over sentiment-augmented knowledge graphs towards accurate and explainable recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 784–793.

[141] Apurva Pathak, Kshitiz Gupta, and Julian McAuley. 2017. Generating and personalizing bundle recommendations on steam. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1073–1076.

[142] 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.

[143] 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.

[144] 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.

[145] 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.

[146] 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.

[147] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. 2019. Fairwalk: Towards fair graph embedding. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 3289–3295.

[148] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the International Conference on Data Mining*. IEEE, 995–1000.

[149] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. 452–461.

[150] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. DropEdge: Towards deep graph convolutional networks on node classification. In *Proceedings of the International Conference on Learning Representations*. 1–18.

[151] Heng-Shiou Sheu and Sheng Li. 2020. Context-aware graph embedding for session-based news recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 657–662.

[152] 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 Sig. Process. Mag.* 30, 3 (2013), 83–98.

[153] 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*. 2219–2228.

[154] 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, 830–831.

[155] 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.

[156] 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.

[157] 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 12th ACM International Conference on Web Search and Data Mining*. 555–563.

[158] Yixin Su, Rui Zhang, Sarah Erfani, and Zhenghua Xu. 2021. Detecting beneficial feature interactions for recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4357–4365.

[159] 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.

[160] 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.

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

[162] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 235–244.

[163] 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*. 297–305.

[164] 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.

[165] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 269–278.

[166] Nava Tintarev and Judith Masthoff. 2007. A survey of explanations in recommender systems. In *Proceedings of the IEEE 23rd International Conference on Data Engineering Workshop*. 801–810.

[167] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Proceedings of the Neural Information Processing Systems Conference*, Vol. 26.

[168] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the International Conference on Learning Representations*. 1–12.

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

[170] 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.

[171] 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*. 417–426.

[172] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *Proceedings of the World Wide Web Conference*. 3307–3313.

[173] Jianling Wang, Kaize Ding, Ziwei Zhu, and James Caverlee. 2021. Session-based recommendation with hypergraph attention networks. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, 82–90.

[174] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, et al. 2019. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315* (2019).

[175] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. 2021. A survey on session-based recommender systems. *Comput. Surv.* 54, 7 (2021), 1–38.

[176] Shoujin Wang, Liang Hu, Yan Wang, Xiangnan He, Quan Z. Sheng, Mehmet A. Orgun, Longbing Cao, Francesco Ricci, and Philip S. Yu. 2021. Graph learning based recommender systems: A review. *arXiv preprint arXiv:2105.06339* (2021).

[177] 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*. 3056–3062.

[178] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2023. Incorporating link prediction into multi-relational item graph modeling for session-based recommendation. *IEEE Trans. Knowl. Data Eng.* 35, 3 (2023), 2683–2696. DOI : 10.1109/TKDE.2021.3111436

[179] Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. 2023. Disentangled representation learning for recommendation. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 1 (2023), 408–424. DOI : 10.1109/TPAMI.2022.3153112

[180] 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.

[181] 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.

[182] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the Web Conference*. 878–887.

[183] 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.

[184] Xiting Wang, Kunpeng Liu, Dongjie Wang, Le Wu, Yanjie Fu, and Xing Xie. 2022. Multi-level recommendation reasoning over knowledge graphs with reinforcement learning. In *Proceedings of the ACM Web Conference*. 2098–2108.

[185] 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.

[186] 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.

[187] 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.

[188] Zhenyi Wang, Huan Zhao, and Chuan Shi. 2022. Profiling the design space for graph neural networks based collaborative filtering. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 1109–1119.

[189] Zhaobo Wang, Yanmin Zhu, Qiaomei Zhang, Haobin Liu, Chunyang Wang, and Tong Liu. 2022. Graph-enhanced spatial-temporal network for next poi recommendation. *ACM Trans. Knowl. Discov. Data* 16, 6 (2022), 21.

[190] 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.

[191] Zhaobo Wang, Yanmin Zhu, Qiaomei Zhang, Haobing Liu, Chunyang Wang, and Tong Liu. 2022. Graph-enhanced spatial-temporal network for next POI recommendation. *ACM Trans. Knowl. Discov. Data* 16, 6 (2022), 21.

[192] 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.

[193] Jiancan Wu, Xiangnan He, Xiang Wang, Qifan Wang, Weijian Chen, Jianxun Lian, and Xing Xie. 2022. Graph convolution machine for context-aware recommender system. *Front. Comput. Sci.* 16, 6 (2022), 1–12.

[194] 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.

[195] 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*. 2198–2208.

[196] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Trans. Knowl. Data Eng.* (2022), 1–1.

[197] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2022. DifNet++: A neural influence and interest diffusion network for social recommendation. *IEEE Trans. Knowl. Data Eng.* 34, 10 (2022), 4753–4766. DOI:10.1109/TKDE.2020.3048414

[198] 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.

[199] Le Wu, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2021. Collaborative neural social recommendation. *IEEE Trans. Syst. Man Cyber.: Syst.* 51, 1 (2021), 464–476.

[200] 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 *Proceedings of the International Joint Conference on Artificial Intelligence*. 1632–1638.

[201] 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 *Proceedings of the World Wide Web Conference*. 2091–2102.

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

[203] 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.

[204] Yuxia Wu, Lizi Liao, Gangyi Zhang, Wenqiang Lei, Guoshuai Zhao, Xueming Qian, and Tat-Seng Chua. 2022. State graph reasoning for multimodal conversational recommendation. *IEEE Trans. Multim.* (2022), 1–1.

[205] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 1 (2020), 4–24.

[206] 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 *Proceedings of the 37th International Conference on Data Engineering*. IEEE, 1931–1936.

[207] 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.

[208] 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.

[209] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. 2021. Self-supervised graph co-training for session-based recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2180–2190.

[210] 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.

[211] 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*. 285–294.

[212] 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. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3786–3794.

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

[214] Ruobing Xie, Qi Liu, Shukai Liu, Ziwei Zhang, Peng Cui, Bo Zhang, and Leyu Lin. 2022. Improving accuracy and diversity in matching of recommendation with diversified preference network. *IEEE Trans. Big Data* 8, 4 (2022), 955–967.

[215] 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 *Proceedings of the International Conference on International Joint Conferences on Artificial Intelligence*, Vol. 19. 3940–3946.

[216] 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.

[217] 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 *Proceedings of the IEEE International Conference on Data Mining*. IEEE, 701–710.

[218] Zhen Yang, Ming Ding, Xu Zou, Jie Tang, Bin Xu, Chang Zhou, and Hongxia Yang. 2022. Region or global a principle for negative sampling in graph-based recommendation. *IEEE Trans. Knowl. Data Eng.* (2022), 1–1.

[219] Zuoxi Yang and Shoubin Dong. 2020. HAGERec: Hierarchical attention graph convolutional network incorporating knowledge graph for explainable recommendation. *Knowl.-based Syst.* 204 (2020), 106194.

[220] 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.

[221] 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).

[222] 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.

[223] 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.

[224] 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.

[225] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-aware self-supervised tri-training for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2084–2092.

[226] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. 2022. Enhance social recommendation with adversarial graph convolutional networks. *IEEE Trans. Knowl. Data Eng.* 34, 8 (2023), 3727–3739. DOI : 10.1109/TKDE.2020.3033673

[227] 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*. 413–424.

[228] Wenhui Yu, Zixin Zhang, and Zheng Qin. 2022. Low-pass graph convolutional network for recommendation. *Proc. AAAI Conf. Artif. Intell.* 36, 8 (2022), 8954–8961.

[229] 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.

[230] Jun Zhang, Chen Gao, Depeng Jin, and Yong Li. 2021. Group-buying recommendation for social e-commerce. In *Proceedings of the IEEE 37th International Conference on Data Engineering*. 1536–1547.

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

[232] 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*. 4264–4270.

[233] Mengqi Zhang, Shu Wu, Meng Gao, Xin Jiang, Ke Xu, and Liang Wang. 2022. Personalized graph neural networks with attention mechanism for session-aware recommendation. *IEEE Trans. Knowl. Data Eng.* 34, 8 (2022), 3946–3957.

[234] Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Trans. Knowl. Data Eng.* (2022), 1–1.

[235] 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.

[236] Yongfeng Zhang and Xu Chen. 2020. Explainable recommendation: A survey and new perspectives. *Found. Trendső Inf. Retr.* 14, 1 (2020), 1–101.

[237] 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.

[238] 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.

[239] Kai Zhao, Yukun Zheng, Tao Zhuang, Xiang Li, and Xiaoyi Zeng. 2022. Joint learning of e-commerce search and recommendation with a unified graph neural network. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 1461–1469.

[240] Sen Zhao, Wei Wei, Ding Zou, and Xianling Mao. 2022. Multi-view intent disentangle graph networks for bundle recommendation. *Proc. AAAI Conf. Artif. Intell.* 36, 4 (2022), 4379–4387.

[241] Ruiqi Zheng, Liang Qu, Bin Cui, Yuhui Shi, and Hongzhi Yin. 2022. AutoML for deep recommender systems: A survey. *arXiv preprint arXiv:2203.13922* (2022).

[242] 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*. 401–412.

[243] Yu Zheng, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2023. Incorporating price into recommendation with graph convolutional networks. *IEEE Trans. Knowl. Data Eng.* 34, 8 (2022), 3727–3739. DOI : 10.1109/TKDE.2020.3033673

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

[245] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. 2020. DGTN: Dual-channel graph transition network for session-based recommendation. In *Proceedings of the International Conference on Data Mining Workshops*. IEEE, 236–242.

[246] 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*. 1284–1295.

[247] Zhi Zheng, Chao Wang, Tong Xu, Dazhong Shen, Penggang Qin, Xiangyu Zhao, Baoxing Huai, Xian Wu, and Enhong Chen. 2023. Interaction-aware drug package recommendation via policy gradient. *ACM Trans. Inf. Syst.* 41, 1 (2023), 1046–8188.

[248] 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.

[249] Huachi Zhou, Qiaoyu Tan, Xiao Huang, Kaixiong Zhou, and Xiaoling Wang. 2021. Temporal augmented graph neural networks for session-based recommendations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1798–1802.

[250] 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.

[251] Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, and Xia Hu. 2020. Towards deeper graph neural networks with differentiable group normalization. *Adv. Neural Inf. Process. Syst.* 33 (2020), 1–12.

[252] 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. *Proc. VLDB Endow.* 12, 12 (2019), 2094–2105.

[253] Tianyu Zhu, Leilei Sun, and Guoqing Chen. 2023. Graph-based embedding smoothing for sequential recommendation. *IEEE Trans. Knowl. Data Eng.* 35, 1 (2023), 496–508. DOI : 10.1109/TKDE.2021.3073411

[254] 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.

[255] Tao Zhuang, Wenwu Ou, and Zhirong Wang. 2018. Globally optimized mutual influence aware ranking in e-commerce search. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence.* 3725–3731.