A Survey on Neural Recommendation: From Collaborative Filtering to Content and Context Enriched Recommendation

Le Wu *Member, IEEE*, Xiangnan He *Member, IEEE*, Xiang Wang *Member, IEEE*,
Kun Zhang *Member, IEEE*, and Meng Wang, *Fellow, IEEE*

Abstract—Influenced by the stunning success of deep learning in computer vision and language understanding, research in recommendation has shifted to inventing new recommender models based on neural networks. In recent years, we have witnessed significant progress in developing neural recommender models, which generalize and surpass traditional recommender models owing to the strong representation power of neural networks. In this survey paper, we conduct a systematic review on neural recommender models, aiming to summarize the field to facilitate future progress. Distinct from existing surveys that categorize existing methods based on the taxonomy of deep learning techniques, we instead summarize the field from the perspective of recommendation modeling, which could be more instructive to researchers and practitioners working on recommender systems. Specifically, we divide the work into three types based on the data they used for recommendation modeling: 1) collaborative filtering models, which leverage the key source of user-item interaction data; 2) content enriched models, which additionally utilize the side information associated with users and items, like user profile and item knowledge graph; and 3) context enriched models, which account for the contextual information associated with an interaction, such as time, location, and the past interactions. After reviewing representative works for each type, we finally discuss some promising directions in this field, including benchmarking recommender systems, graph reasoning based recommendation models, and explainable and fair recommendations for social good.

Index Terms—Recommendation Survey, Deep Learning, Neural Networks, Neural Recommendation Models

1 Introduction

INFORMATION overload is an increasing problem in people's every life due to the proliferation of the Internet. On a par with search engine, recommender system serves as an effective solution to alleviate the information overload issue, to facilitate users seeking desired information, and to increase the traffic and revenue of service providers. It has been used in a wide range of applications, such as E-commerce, social media sites, news portals, App stores, digital libraries, and so on. It is one of the most ubiquitous user-centric artificial intelligence applications in modern information systems.

The research in recommendation can be dated back to the 1990s [1], in the age the early work developed many heuristics for content-based and Collaborative Filtering (CF) [2]. Popularized by the Netflix challenge, Matrix Factorization (MF) [3] later becomes the mainstream recommender model for a long time (from 2008 until 2016) [4], [5]. However, the linear nature of factorization models makes them less effective when dealing with large and complex data, e..g, the complex user-item interactions, and the items may contain complex semantics (e.g., texts and images) that require a thorough understanding of them. Around the same time in mid-2010s, the rise of deep neural networks in machine learning (aka., Deep Learning) has revolutionized several areas including speech recognition, computer vision, and natural language processing [6]. The great success of deep learning stems from the considerable expressiveness of neural networks, which are particularly advantageous for learning from large data with complicated patterns. This naturally brings new opportunities to advance the recommendation technologies. And not surprisingly, there emerges a lot of work on developing neural network approaches to recommender systems in the past several years. In this work, we aim to provide a systematic review on the recommender models that use neural networks — referred to as "*neural recommender models*". This is the most thriving topic in current recommendation research, not only has many exciting progress in recent years, but also shows the potential to be the technical foundations of the next-generation recommender systems.

Differences with Existing Surveys. Adomavicius et al. presented an overview of recommender systems at early stages before 2005 [2]. Shi et al. [7] reviewed methods that integrated side information such as attributes and contexts into collaborative filtering, and Khan et al. [8] focused on cross-domain recommendation that integrated data from multiple domains. All these surveys are not emphasized on neural network techniques

- *L. Wu, and M Wang are with Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Hefei, Anhui 230029, China, and Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, Anhui 230088 (email: lewu.ustc, eric.mengwang@gmail.com).*
- *K. Zhang is with Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Hefei, Anhui 230029, China.(email:zhang1028kun@gmail.com).*
- *X. He is with University of Science and Technology of China, Hefei 230026, China. (email: xiangnanhe@gmail.com).*
- *X. Wang is with National University of Singapore, Singapore. (email: xiangwang@u.nus.edu).*

and are published at the time that deep learning has not become prevalent in recommendation community. Some recent surveys targeted at recommendation methods that are built with Deep Learning (DL). They followed the taxonomy of DL techniques to categorize the recommendation literature. Recently, researchers surveyed specific topics of recommender systems, such as explainable recommendation [9], knowledge graph enhanced recommendation [10], and recommendation models that leverage multimedia data. These surveys focused on specific directions of recommendation, with both the classical approaches and how the neural network based models advance these fields. We differ from these surveys as we aim to give a general overview of typical recommendation tasks in this survey. Therefore, instead of focusing on a specific recommendation direction, our survey covers the most typical recommendation scenarios, including CF, content based recommendation and context aware recommendation.

Scope and Organization of This Survey. We focus on the general task of item recommendation, i.e., recommending items to users, and leave out the discussion of other recommendation tasks like recommending items for a group of users (aka., group recommendation), recommending tags to items (aka., tag recommendation [11]), domain specific recommendation (e.g., education recommendation and fashion recommendation). Moreover, we concentrate on the work that leverages the data of a single domain, leaving out the discussion on cross-domain recommendation [8]. Our goal is to provide a comprehensive survey of the general item recommendation in a single domain, and help young researchers to grasp the main research directions in this field.

The organization of this survey is structured as follows. In Section 2, we review *collaborative filtering models* that use the ID and interaction history for modeling. In Section 3, we review the models that integrate the side information of users and items into recommendation, such as user profiles and social networks, item attributes and knowledge graphs. We term them as *content-enriched models*, which naturally extend CF by integrating the side information. In Section 4, we review the models that account for contextual information. The contextual data is associated with each user-item interaction such as time, location, and the past interaction sequence. The *context-aware models* make predictions based on the context data. Due to page limits, we focus on temporal information, which is one of the most common contextual data. Finally, we conclude the survey and propose some promising future directions.

2 Collaborative Filtering Models

The concept of CF stems from the idea that leveraging the collaborative behaviors of all users for predicting the behavior of the target user. Early approaches directly implement this idea by calculating the behavior similarity of users (user-based CF) or of items (item-based CF) with memory based models. Later on, matrix factorization based models become prevalent which implement the CF idea by collectively finding the latent spaces that encode user-item interaction matrix [3], [12]. These models can predict users' preferences to some extent. However, they suffered from limited prediction power due to the conflict between users' complex preferences and the simply linear modeling ability. Given the expressive complex modeling power of neural networks, the current solutions for neural CF can be summarized into two categories: representation modeling of users and items, and user-item interaction modeling given the representations.

2.1 Representation Learning

Let U and V denote users and items in CF, with $\mathbf{R} \in \mathbb{R}^{M \times N}$ is the user-item interaction behavior matrix. The general objective of representation learning in CF is to learn a user embedding matrix $\mathbf{P}$ and an item embedding $\mathbf{Q}$, with $\mathbf{p}_u$ and $\mathbf{q}_i$ denote the representation parameters for user u and item i, respectively.

In fact, as each user performs limited behavior compared to the large item set, a key challenge that lies in CF is the sparsity of the user-item interaction behavior for accurate user and item embedding learning. Different kinds of representation learning models vary in the input data for representation learning, as well as the representation modeling techniques given the input data. We divide this section into three categories: history behavior aggregation enhanced models, autoencoder based models, and graph learning approaches. For ease of explanation, we list the typical representation learning models in Table 1.

2.1.1 History Behavior Attention Aggregation Models

By taking the one-hot User ID (UID), and one-hot Item ID (IID) as input, classical latent factor models associate each UID u and IID i with a free embedding vector of $\mathbf{p}_u$ and $\mathbf{q}_i$ [3], [12]. Instead of modeling users with free embeddings, researchers further proposed borrowing users' historical behavior for better user representation modeling. E.g., Factored Item Similarity Model (FISM) pools the interacted item embeddings as a user representation vector [13], and SVD++ [17] adds UID embedding $\mathbf{p}_u$ with the interaction history embedding (i.e., the FISM user representation) as the final user representation. These models relied on simple linear matrix factorization, and used heuristics or equal weights for the interaction history aggregation.

However, the previous history behavior aggregation techniques are not perfect in practice, as different historical items contribute differently to model a user's preference. To address this limitation, some researchers integrate neural attention mechanism into history representation learning [18], [19], [35]. One representative work is Attentive Collaborative Filtering (ACF) [19], which assigns each interacted item with a user-aware attentive weight to indicate its importance to user representation:

$$\hat{r}_{ui} = (\mathbf{p}_u + \sum_{j \in R_u} \alpha(u,j)\mathbf{q}'_j)^T \mathbf{q}_i, \tag{1}$$

where $\mathbf{p}_u$ is the ID embedding of user u, R_u denotes the items that u has interacted with. $\alpha(u,j)$ is the attentive weight defined as:

$$\alpha(u,j) = \frac{\exp(F(\mathbf{p}_u, \mathbf{q}'_j))}{\sum_{j' \in R_u} \exp(F(\mathbf{p}_u, \mathbf{q}'_{j'}))}, \tag{2}$$

TABLE 1: Summarization of representation learning approaches for CF

Type	Input		Representation Modeling		Models
	User	Item	User Representation	Item Representation	
Classical Matrix Factorization	UID	IID	Free Embed.	Free Embed.	BPR [12], MF [3] et al.
	Interacted items	IID	Free embed.+Heuristic Agg.	Free Embed.	FISM [13], PMLAM [14], pQCF [15], FAWMF [16]
	Interacted Items+ UID	IID	Free Embed+Heuristic Agg.	Free Embed.	SVD++ [17]
History Attention	Interacted Items	ID	Free Embed+ Attention Agg.	Free Embed.	NAIS [18]
	Interacted Items+ UID	IID	Free Embed. + Attention Agg.	Free Embed.	ACF [19]
Autoencoder Models	***	Interacted Items	***	Non-linear Encoder	AutoRec [20], CDAE [21], Mult-VAE [22] et al.
	Interacted Items	Interacted Users	Non-linear encoder	Non-linear encoder	REAP [23], CE-VNCF [24], SW-DAE [25]
Graph Learning	UID+ Graph	IID+Graph	GNN		GC-MC [26], NGCF [27], SpectralCF [28], NIA-GCN [29], BGCF [30], DGCF [31] et al.
	UID+ Graph	IID+ Graph	Simplified GNN		LR-GCCF [32], LightGCN [33], DHCF [34] et al.

where $F(\;;\;)$ is a function that can be implemented as a MLP or simply inner product.

In practice, the influence of a historical item can be dependent on the target item, e.g., the purchase of a phone case is more related to the previous purchase of phone, while the purchase of a pant could be more related to the previous purchase of a shirt. As such, it may be beneficial to have dynamic user representation when considering the prediction on different target items. To this end, the Neural Attentive Item Similarity model (NAIS) model [18] revises the attention mechanism to be target item-aware:

$$\hat{r}_{ui} = (\sum_{j \in R_u} (i;j)\, q_j^{0})^T q_i$$
$$(i;j) = \frac{\exp(F(q_i; q_j^0))}{[\sum_{k \in R_u} \exp(F(q_i; q_k^0))]}; \quad (3)$$

where $(i;j)$ denotes the contribution of historical item j to user representation when predicting a user's preference on target item i. is a hyper-parameter between 0 and 1 (e.g., 0.5), for smoothing the interaction histories of different lengths. Similar attention mechanisms have adopted for representation learning from interaction history, e.g., the Deep Item-based CF model(DeepICF) [35] and Deep Interest Network (DIN) [36]. As such, interaction history contains more information than single user ID and is a suitable choice for representation learning.

2.1.2 Autoencoder based Representation Learning

Autoencoder based models take the incomplete user-item matrix as input, and learn a hidden representation of each instance with an encoder, and further with a decoder part that reconstructs the input based on the hidden representation. By treating each user's historical records as input, the autoencoder based models learn each user's latent representation with a complex encode neural network, and feed the learned user representation into a decoder network to output the predicted preference of each user. An alternative approach is to take each item's rating records from all users as input, and learn the item's latent representation to reconstruct the predicted preference of each item from all users [20], [21]. Extensions on this line of works can be classified into two categories. The first category leveraged autoencoder variants, and injected denoising autoencoders [21], variational autoencoders [22] into CF. These models can be seen as using complex deep learning techniques for learning either user or item encoders. The second category exploited the duality of users and items in autoencoders, and designed two parallel encoders to learn the user and item representations, and then also use inner product to model users' preferences to items [23]. It is worth pointing out the autoencoder based CF approaches can also be classified as extensions of the historical behavior attention based models, as these approaches adopt deep neural networks for aggregating historical behavior.

2.1.3 Graph based Representation Learning

The CF effects are reflected in the interaction histories of multiple users. As such, using collective interaction histories has the potential to improve the representation quality. From the perspective of user-item interaction graph, the individual interaction history is equivalent to the first-order connectivity of the user. Thus, a natural extension is to mine the higher-order connectivities from the user-item graph structure. E.g., the second-order connectivity of a user consists of similar users who have co-interacted with the same items. Fortunately, with the success of Graph Neural Networks (GNNs) for modeling graph structure data in the machine learning community [37], many works have been proposed how to model the user-item bipartite graph structure for neural graph based representation learning. Given the user-item bipartite graph, let P^0 and Q^0) denote the free user latent matrix and item latent matrix as many classical latent factor based models, i.e., the0^{th} -order user and item embedding. These neural graph based models iteratively update the $(l+1)^{th}$ -order user (item) emebedding as an aggregation of the l^{th} -order item (user) embedding. E.g., each useru's updated embedding $p_u^{(l+1)}$ is calculated as:

$$p_u^{(l+1)} = (W^l[a_u^l; a_u^{(l+1)}]); \quad (4)$$
$$a_u^{(l+1)} = Agg(q_j^{(l\;\;1)} | j \in R_u); \quad (5)$$

where q_j^l is item j's representation at l^{th} layer, R_u denotes the items that connect to user u in the user-item bipartite graph. $a_u^{(l+1)}$ is the aggregation of connected items' representations in the l^{th} layer, W^l is an embedding transformation matrix that needs to be learned, and $()$ is an activation function. After that, each user's (item's) final embedding can be seen as combining each entity's embedding at each layer.

The above steps can be seen as embedding propagation in the user-item bipartite graph. With a predefined layer L, the up to L^{th} order sub graph structure is directly encoded in the user and item embedding representation step. E.g., SpectralCF utilized the spectral graph convolutions for CF [28]. GC-MC [26] and NGCF [27] modeled the graph convolutions of user-item behavior in the original space, and are more effective and efficient in practice. Very recently, researchers argued that these neural graph based CF models differ from the classical GNNs as CF models do not contain any user or item features, and directly borrowing complex steps such as embedding transformation, and non-linear activations in GNNs may not be a good choice. Simplified neural graph CF models, including LR-GCCF [32], and LightGCN [33] have been proposed, which eliminate unnecessary deep learning operations. These simplified neural graph based models show superior performance in practice without the need of carefully chosen activation functions.

2.2 Interaction Modeling

Let p_u and q_i denote the learned embeddings of users and items from representation models, this component aims at interaction function modeling that estimates the user's preference towards the target item based on their representations. In the following, we describe how to model users' predicted preference, denoted as $\hat{r}_{ui}$ based on the learned embeddings. For ease of explanation, as shown in Table 2, we summarize three main categories for interaction modeling: classical inner product based approaches, distance based modeling and neural network based approaches.

Most previous recommendation models relied on the inner product between user embedding and item embedding to estimate the user-item pair score as: $\hat{r}_{ui} = p_u^{\top} q_i = \sum_{f=1}^{d} p_{uf} q_{if}$. Despite its great success and simplicity, prior efforts suggest that simply conducting inner product would have two major limitations. First, the triangle inequality is violated [38]. That is, inner product only encourages the representations of users and historical items to be similar, but lacks guarantees for the similarity propagation between user-user and item-item relationships. Second, it models the linear interaction, and may fail to capture the complex relationships between users and items [41].

2.2.1 Distance based Metrics

In order to solve the first issue, a line of research [38], [39], [40] borrows ideas from the translation principles and uses distance metric as the interaction function. The inherent triangle inequality assumption plays an important role in helping capture the underlying relationships among users and items. For instance, if user u tends to purchase items i and j, the representations of i and j should be close in the latent space.

Towards this end, CML [38] minimizes the distance d_{ui} between each user-item interaction $< u; i >$ in Euclidean space as: $d_{ui} = \|p_u - q_i\|_2^2$. Instead of minimizing the distance between each observed user-item pair, TransRec exploits the translation principle to model the sequential behaviors of users [39]. In particular, the representation of user u is treated as the translation vector between the representations of the items i and the item j to visit next, namely, $q_j + p_u \quad q_i$.

Distinct from CML that uses simple metric learning that assumes each user's embedding is equally close to every item embedding she likes, LRML introduces the relation vectors r to capture the relationships between user and item pairs [40] . More formally, the score function is defined as:

$$s_{ui} = \|p_u + e - q_i\|_F^2 ; \qquad (6)$$

where the relation vector $e \in \mathbb{R}^d$ is constructed using a neural attention mechanism over a memory matrix M. $M \in \mathbb{R}^{m \times d}$ is the trainable memory module, hence E is the attentive sum of m memory slots. As a result, the relation vectors not only ensure the triangle inequality, but also achieve better representation ability.

2.2.2 Neural network based Metrics

Distinct from the foregoing that employs linear the metrics, recent works adopt a diverse array of neural architectures, spanning from MLP, Convolutional Neural Network (CNN), and AE as the main building block to mine complex and nonlinear patterns of user-item interactions.

Researchers made attempts to replace similarity modeling between users and items with MLPs, as MLPs are general function approximators to model the any complex continuous function. NCF is proposed to model the interaction function between each user-item pair with MLPs as: $\hat{r}_{ui} = f_{MLP}(p_u || q_i)$. Besides, NCF also incorporates a generic MF component into the interaction modeling, thereby making use of both linearity of MF and non-linearity of MLP to enhance recommendation quality.

Researchers also proposed to leverage CNN based architecture for interaction modeling. These kinds of models first generate interaction maps via outer product of user and item embeddings, explicitly capturing the pairwise correlations between embedding dimensions [42], [43]. These CNN based CF focuses on high order correlations among representation dimensions. However, such improvements on performance come at the cost of increasing model complexity and time cost.

Besides, a line of research exploits AEs to fulfill the blanks of the user-item interaction matrix directly in the decoder part [20], [21], [22], [23], [44], [45], [46]. As the encoder and decoder can be implemented via deep neural networks, such stacks of nonlinear transformations give the recommenders more capacity to model the user representation from complex combinations of all historically interacted items.

3 Content-enriched Recommendation

Besides the general user-item interaction information, recommendation problems are often accompanied with auxiliary data. The auxiliary data could be classified into two categories: content based information and context-aware data. Specifically, the first category of content information is associated with users and items, including general user and item features, textual content (a.k.a, item tags, item textual descriptions and users' reviews for items), multimedia descriptions (a.k.a, images, videos, and audio information), user social networks, and knowledge graphs. In contrast, contextual information shows the environment when users make item decisions, which usually denotes descriptions that beyond users and items [2]. Contextual information includes time, location, and specific data that are collected from sensors (such as speed, and

TABLE 2: Interaction modeling techniques

Category	Modeling Idea	Models
Inner Product	$\hat{r}_{ui} = p_u^{>} q_i$	Most models
Distance Modeling	Euclidean distance $d_{ui} = kp_u \quad q_i k_2^2$	CML [38]
	Nearby translation $\hat{d}_{ui} = \ _j \quad d(q_j + p_u; q_i)$	TransRec [39]
	Memory enhanced Translation $\hat{d}_{ui} = kp_u + E \quad q_i k_2^2$	LRML [40]
Neural Networks	$\hat{r}_{ui} = MLP\ (p_u jj q_i)$	NCF [41] et al.
	$\hat{r}_{ui} = CNN\ (p_u \quad q_i)$	ONCF [42] et al.
	Autoencoder based reconstruction $kr_i \quad dec(enc(r_i)) k_2^2$	AutoRec [20], CDAE [21] et al.

weather), and so on. Due to page limits, we discuss the most typical contextual data: temporal data. In the following of the two sections, we would give a detailed summary of the content-enriched recommendation and context-aware recommendation. For the content-enriched recommendation, we classify the related works into five categories based on the available content information: the general features of users and items, the textual content information, the multimedia information, social networks and knowledge graphs.

3.1 Modeling General Feature Interactions

Factorization Machine (FM) provides an intuitive idea of feature interaction modeling [47]. As the features are usually sparse, FM first embed each feature i into a latent embedding v_i, and then model second-order interaction of any two feature instances with values of x_i and x_j as: $v_i^T v_j x_i x_j$. Naturally, FM models the second-order interactions, and reduces the parameter size of computing similarity of any two features with embedding based models. FM has been extended to field-aware FM by expanding each feature with several latent embeddings based on the field aware property [48], or higher-order FMs by directly expanding 2-order interactions with all feature interactions [59]. Despite the ability to model higher-order interactions, these models suffer from noisy feature interactions in the modeling process.

Researchers have explored the possibility of adopting neural models to automatically discover complex higher-order feature interactions for CTR prediction and recommendation. As shown in Table 3, besides the FM based approaches, currently the related works on this topic can be classified into three categories: implicit MLP structures and explicit up to K-th order modeling, and tree enhanced models.

MLP based High Order Modeling. As the feature interactions are hidden, researchers proposed to first embed each feature with an embedding layer, and then exploit MLPs to discover high order correlations. This category can be seen as modeling feature interactions in an implicit way as MLPs are black-box approaches, and we do not know what kind of feature interactions from the output of the MLP structure models. Since MLPs suffer from training difficulties, some researchers proposed pretraining techniques [50]. Others injected specific structures in MLPs for better capturing feature interactions. DeepCrossing designed residual structures to add back the original input after every two layers of MLPs [52]. The NFM architecture has a proposed bi-interaction operation before MLP layers [49]. PNN modeled both the bit-wise interactions of feature embedding interactions and vector-wise feature interactions [51]. Besides the complex high order interactions, another effective approach is to combine the MLP based high order modeling with the classical linear models [54], [55].

Cross Network for K-th Order Modeling. The cross network differs from the MLP based approaches with a carefully designed cross network operation, such that a K^{th} layer cross network models the up to K^{th} order feature interactions. The k^{th} hidden layer output x_k is calculated by the cross operation as: $x_k = x_0 x_{k\ 1} w_k + b_k + x_{k\ 1}$ [56]. Instead of operating cross operations at a bitwise level, xDeepFM applies cross interactions at the vector-wise level explicitly [57]. These kinds of models are able to learn bounded-degree feature interactions.

Tree Enhanced Modeling. As trees can naturally show cross feature interactions, researchers incorporated trees as a proxy for recommendation with cross feature explanation. Specifically, TEM [58] first utilizes decision trees to extract high order interaction of features in the form of cross features, and then input embeddings of cross features into an attentive model to perform prediction. As a result, the depth of the decision tree determines the maximum degree of feature interactions. Furthermore, by combining embedding and tree based models seamlessly, TEM is able to unify their strengths — strong representation ability and explainability.

3.2 Modeling Textual Content

Neural network technique has revolutionized Natural Language Processing (NLP) [60], [61]. These neural NLP models enable multi-level automatic representation learning of textual content, and can be combined in the recommendation framework for better user and item semantic embedding learning. Given the above neural based NLP models, we discuss some typical textual enhanced recommendation models based on the above techniques. Textual content input for recommendation could be classified into two categories: the first category is the content descriptions associated with either items or users, such as the abstract of an article, or the content descriptions of a user. The second category links a user-item pair, such as users annotating tags to items, or writing reviews for products. For the second category, most models summarize the associated content with each user, and each item [62], [63]. Under such a situation, the second category of content information degenerates to the first category. In the following, we do not distinguish the input content data types, and summarize the related works for modeling contextual content into the following categories:

TABLE 3: Classification of modeling feature-enhanced CF

Category	Modeling Idea	Models
Second order	Model second order correlations with embedding based similarity	FM [47], FFM [48]
MLP based higher order	Design better initialization techniques to facilitate MLP modeling	NFM [49], FNN [50], PNN [51], DeepCrossing [52], [53]
	Combine deep and shallow features	Wide&Deep [54], DeepFM [55]
Up to K^{th} order modeling	Deep cross network structure for defined order depth	Deep& Cross [56], xDeepFM [57]
Tree structure	Tree enhanced embedding for attentive cross feature aggregation	TEM [58]

autoencoder based models, word embeddings, attention models, and text explanations for recommendation.

Autoencoder based Models. By treating item content as raw features, such as bag-of-words and item tag representations, these models use autoencoders and their variants to learn the bottleneck hidden content representations of items. Different autoencoder based models vary in the detailed choices of autoencoder implementations, and how to associate the bottleneck content representations from autoencoders and the hidden latent factors in collaborative filtering for better user (item) modeling [23], [46], [64], [65], [66], [67], [68], [69], [70], [71]. E.g., Collaborative Deep Learning (CDL) [64] is proposed to simultaneously learn each item i's embedding q_i as a combination of two parts: a hidden representation from the item content x_i with a stacked denoising autoencoder and an auxiliary embedding that is not encoded in the item content as:

$$q_i = f_e(x_i) + {}_i; \quad {}_i \quad N(0; {}^2) \tag{7}$$

where $f_e(x)$ transforms raw content input into a bottleneck hidden vector with an autoencoder. ${}_i$ is a free item latent vector that is not captured in the item content, which is similar as many classical latent factor based CF models. In the model optimization process, the objective function is to simultaneously optimize the rating based loss from users' historical behavior and the content-reconstruction loss from the autoencoder:

$$L = L_R(R; \hat{R}) + \quad L_X(X; f_d(f_e(X))); \tag{8}$$

where is a balance parameter that measures the relative weight between the two loss terms. In the above optimization function, R is the user-item rating matrix and $\hat{R}$ denotes the predicted rating. Similarly, X is the item content input and $f_d(f_e(X))$ is an reconstructed content from an autoencoder that encodes the item content into a bottleneck representation f_e, and then reconstructs it with a decoder f_d.

Following this basic autoencoder based recommendation model, some studies proposed improvements to consider the uniqueness of the content information. E.g., as the basic autoencoders take bag-of-words as input without any order information, a collaborative recurrent autoencoder is proposed to jointly model the order-aware reconstruction of item content and the rating information in CF [65]. Instead of learning a deterministic vector representation of the item content, a Collaborative Variational AutoEncoder (CVAE) is proposed to simultaneously recover the rating matrix and the side content information with a variational autoencoder [67]. Researchers also proposed to leverage the item neighbor information from item content to better represent the bottleneck representation of the item [72]. For some recommendation scenarios, items are also associated with category information. A denoising autoencoder with weak supervision is proposed to learn the distributed representation vector of each item, where items in the same category are more similar to items in different categories [73]. Besides, as both users and items could be associated with content information, dual autoencoder based recommendation models have been proposed, with an autoencoder learns the user content bottleneck representation and its dual autoencoder learns the item content bottleneck representation [23], [46], [66], [74].

Leveraging Word Embeddings for Recommendation. Autoencoders provide general neural solutions for unsupervised feature learning, which do not take the uniqueness of text input into consideration. Recently, researchers proposed to leverage word embedding techniques for better content recommendation [62], [75], [76], [77], [78], [79], [80], [81]. With the success of TextCNN [82], a Convolutional Matrix Factorization (ConvMF) is proposed to integrate CNN into probabilistic matrix factorization [75]. Let x_i denote the text input of item i. The item latent embedding matrix Q is then represented as a Gaussian distribution that centers around its embedding representation as:

$$p(QjW; X; {}^2) = \prod_{i=1}^{jVj} N(q_i jTextCNN(W; x_i); {}^2); \tag{9}$$

where W is the parameters in TextCNN module. Besides using CNN based models for document representation, researchers also employed various state-of-the-art content embedding techniques, such as RNNs for item content representation [83].

Reviews widely appear in recommendation applications and are natural forms for users to express feelings about items. Given user's rating records and associated reviews, most review based recommendation algorithms aggregate historical review text of users (items) as user content input $D(u)$ (item content input $D(i)$). Deep Cooperative Neural Network (DeepCoNN) [62] is a state-of-the-art deep model for review based recommendation. DeepCoNN consists of two parallel TextCNNs for content modeling: one focuses on learning user behaviors by exploiting review content $D(u)$ written by user u, and the other one learns item embedding from the reviews $D(u)$ written for item i. After that, a factorization machine is proposed to learn the interaction between user and item latent vectors. Specifically, DeepCoNN can be formulated as:

$$\hat{r}_{ui} = FM(TextCNN(D_u); TextCNN(D_i)): \tag{10}$$

Many studies have empirically found that the most predictive power of review text comes from the particular review of the target user to the target item. As the associated reviews of a user-item pair are not available in the test stage, TransNet is proposed to tackle the situation when the target review information is not available [76]. TransNet has a source network of DeppCoNN that does not include the joint review rev_{AB}, and a target network that models the joint review of the current user-item pair. Therefore, the target network could approximate the predicted review $\hat{rev}_{AB}$ for the test user-item pair even when users do not give reviews to items.

Attention Models. Attention mechanism has also been widely used in content enriched recommender systems. Given textual descriptions of an item, attention based models have been proposed to assign attentive weights to different pieces of content, such that informative elements are automatically selected for item content representation [84], [85], [86], [87], [88], [89], [90], [91]. E.g., given a tweet, the attention based CNN learns the trigger words in the tweet for better hashtag recommendation [84]. With the historical rated items of a user, an attention model is proposed to selectively aggregate content representations of each historical item for user content preference embedding modeling [92], [93], [94]. Given user (item) collaborative embeddings, and content based embeddings, attention networks have also been designed to capture the correlation and alignment between these two kinds of data sources [91], [95]. Researchers have also proposed a co-evolutionary topical attention regularized matrix factorization model, with the user attentive features learned from an attention network that combines the user reviews, and the item attentive features learned from an attention network that combines the item reviews [95]. For review based recommendation, researchers argued that most content based user and item representation models neglected the interaction behavior between user-item pairs, and a dual attention model named DAML is proposed to learn the mutual enhanced user and item representations [96]. As item content sometimes is presented in multi-view forms (e.g., title, body, keywords and so on), multi-view attention networks are applied to learn unified item representations by aggregating multiple representations from different views [97], [98], [99]. With both the textual descriptions and the image visual information, co-attention is utilized to learn the correlation between the two modalities for better item representation learning [100], [101].

Text Explanations for Recommendation. Instead of improving recommendation accuracy with content input, there is a growing interest of providing text explanations for recommendation. Current solutions for explainable recommendations with text input can be classified into two categories: extraction based models and generation based models.

Extraction based models focus on selecting important text pieces for recommendation explanation. Attention techniques are widely used for extraction based explainable recommendation, with the learned attentive weights empirically show the importance of different elements for model output [97], [102]. After that, the text pieces with larger attentive weights are extracted as recommendation explanations. A neural attentive regression model with review-level explanations is proposed to select useful reviews in the recommendation process. Specifically, the item content representation is composed of an attention network that attentively combines all the related reviews. Therefore, the usefulness of each review could be learned from the attention network [102]. Researchers designed a reinforcement learning model for explainable recommendation given the user reviews. The proposed framework can explain any recommendation model (model-agnostic) and flexibly control the explanation quality based on the application scenario, where the reward of the agent is defined as the quality of the selected reviews [103].

Recently, with the huge success of language generation techniques [104], an alternative approach for explainable recommendation is to generate explanation text for each user [63], [105], [106], [107], [108], [108], [109]. Given both users' rating records and reviews, the key idea of these models is to design an encoder-decoder structure, with the encoder part encodes related embeddings of users and items, and the decoder generates reviews that are similar to the ground truth of the corresponding user-item review text. NRT is a state-of-the-art model that simultaneously predicts ratings and generates reviews [105]. By taking the one-hot user representation and item representation, the encoder part outputs the user latent embedding p_u and item latent embedding q_i, and the review generation process is modeled with an RNN based decoder structure as:

$$\begin{aligned} h^0 &= f(p_u; q_i); \quad h^t = LSTM(h^{t-1}; x^t) \\ \hat{s}_t &= f_d(h^t) \end{aligned} \tag{11}$$

where h^0 is the initial state of the LSTM unit that is a function of the user and item embedddings. x_t is the t-th word of the corresponding user-item review, and h^t represents the hidden state of the LSTM at that time. $f_d(h^t)$ denotes a decoder structure that outputs predicted word $\hat{s}^t$ for review generation. Since we have both the ground truth rating records and the corresponding records of users, the two tasks of rating prediction and review generation can be trained in a multi-task framework.

Researchers have also proposed more advanced encoder-decoder structures for explainable recommendation with text generation techniques. Specifically, the encoder part can encode not only the latent ID information of users and items, as well as the user and item attributes [106], [110], multimodal item data [111], and key review and concept terms that are important with carefully designed attention selector [107].

3.3 Modeling Multimedia Content

With the popularity of multimedia based platforms, the visual content based multimedia contents, e.g., images, videos and music, are the most eye-catching for users. In the following, we introduce related works on modeling multimedia content in recommender systems. For ease of explanation, we summarize the related works on multimedia based recommendation with different kinds of input data in Table 4.

TABLE 4: Multimedia based recommendation models with different kinds of multimedia input

Input	Model Summarization	Models
Image	CNN content based features	ACF [19],VBPR [112],OutfitNet [113]
	Aesthetic based features pretrained from a deep aesthetic network	BDN [114]
	CNN content based features and the style features from feature maps of CNNS	DMF [115]
	Fine grained image attributes	SAERS [116], SNMO [117], AIC [118]
	Co-attention networks for learning enhanced user and image representation	UVCAN [119]
	GNNs to model visual relationships	PinSage [120], HFGN [121], TransGec [122]
Image+ Behavior Time	CNN based temporal content evolution	BDN [114], [123]
Image+Text	Deep fusion networks to learn unified item representation	DMF [115], GraphCAR [124], CKE [125], Transnfcm [126]
	Co-attention networks for learning unified item representation	CoA-CAMN [100], Co-Attention [101]
	Multi-task learning model with detailed image attributes	[127]
	Text generation models by encoding user, item text and image content	VECF [110], KFRCI [128], MRG [111], [129]
Audio	Learning deep audio features	HLDBN [130] [131], [132]
Video	Attention networks to learn video representations from multiple image representations	ACF [19], JIFR [133], AGCN [134]
	GNNs to learn video representation	AGCN [134]
Video+Audio	Deep fusion networks to learn unified item representation	CDML [135], [136]

3.3.1 Modeling Image Information

The current solutions for image recommendation can be categorized into two categories: content based models and hybrid recommendation models. Content based models exploit visual signals for constructing item visual representations, and the user preference is represented in the visual space [101], [115], [116], [117], [124], [137], [138], [139], [140]. In contrast, the hybrid recommendation models alleviate the data sparsity issue in CF with item visual modeling [19], [112], [114], [123], [127], [141], [142].

Image Content based Models. Image content based models are suitable for recommendation scenarios that rely heavily on visual influence (e.g., fashion recommendation) or new items with little user feedback. As visual images are often associated with text descriptions (e.g., tags, titles), researchers designed some unpersonalized recommender systems that suggest tags to images [101], [138]. These models apply CNNs to extract image visual information, and content embedding models to get textual embedding. Then, in order to model the correlation between visual and textual information, these models either project text and images into a same space [115], concatenate representations from different modalities [126] or design co-attention mechanism to better describe items [100], [101].

For personalized image recommendation, a typical solution is to project both users and items in the same visual space, with the item visual space derived from CNNs, and the user's visual preference either modeled by the items they like [137] or a deep neural network that takes the user related profiles as input [124], [139]. Researchers have also argued that CNNs focus on the global item visual representation without fine-grained modeling. Therefore, some sophisticated image semantic understanding models have been proposed to enhance image recommendation performance [115], [116], [117], [118]. E.g., in order to suggest makeups for people, makeup related facial traits are first classified into structured coding. The facial attributes are then fed into a deep learning based recommendation system for personalized makeup synthesis [117]. In some visual based recommendation domains, such as the fashion domain, each product is associated with multiple semantic attributes [116], [118]. To exploit users' semantic preferences for detailed fashion attributes, a semantic attributed explainable recommender system is proposed by projecting both users and items in a fine-grained interpretable semantic attribute space [116].

Hybrid Recommendation Models. Hybrid models utilize both the collaborative signals and the visual content for recommendation, which could alleviate the data sparsity issue in CF and improve recommendation performance. Some researchers proposed to first extract item visual information as features, and the item visual features are fed into factorization machines for recommendation. Instead of the inferior performance induced by the two step learning process, recent studies proposed end-to-end learning frameworks for hybrid visual recommendation [19], [112], [123], [127], [133], [141]. Visual Bayesian Personalized Ranking (VBPR) is one of the first few attempts that leverage the visual content for unified hybrid recommendation [112]. In VBPR, each user (item) is projected into two latent spaces: a visual space that is projected from the CNN based visual features, and a collaborative latent space to capture users' latent preferences. Then, given a user-item pair $(u; i)$ with the associated image x_i, the predicted preference $\hat{r}_{ui}$ is learned by combining users' preferences from two spaces:

$$\hat{r}_{ui} = p_u^T q_i + w_u^T f(CNN(x_i)); \qquad (12)$$

where $f(CNN(x_i))$ denotes the item content representation by transforming items from the original visual space $CNN(x_i)$. In this equation, the first term models the collaborative effect with free user latent vector p_u and item latent vector q_i. The second term models the visual content preference with the item visual embeddings as $f(CNN(x_i)$, and the user visual embedding w_u in the visual space.

Given the basic idea of VBPR, researchers have further introduced the temporal evolution of visual trends in the visual space [123], or the associated location representation of the image [141]. Instead of representing users' preferences into two spaces, the visual content of the item has been leveraged as a regularization term in matrix factorization based models, ensuring that the learned item latent vector of each item is similar to the visual image representation learned from CNNs [123]. Besides learning the CNN content representations for item visual representation, many models have been proposed to consider additional information from the imagery for item visual representation, such as the pretrained aesthetics learned from a deep aesthetic network [114]. As users show time-synchronized comments on video frames, researchers proposed a multi-modal framework to simultaneously predict users' preferences to key frames and generate personalized comments [128]. Compared to review generation models [105], the visual embedding is injected into both the user preference prediction part, as well as each hidden state of the LSTM architecture for better text generation.

Recently, GNNs have shown powerful performance in modeling graph data with heuristic graph convolutional operations [143], [144]. PinSage is one of the first few attempts to apply GNNS for web-scale recommender systems [120]. Given an item-item correlation graph , PinSage takes the node attributes as input, and iteratively generates node embeddings to learn the graph structure with iterative graph convolutions. Researchers also proposed to formulate a heterogeneous graph of users, outfits and items, and performed hierarchical GNNs for personalized outfit recommendation [121].

3.3.2 Video Recommendation

Researchers proposed content-based video recommender systems with rich visual and audio information [135], [136]. Specifically, these proposed models first extracted video features and audio features, and then adopted a neural network to fuse these two kinds of features with early fusion or late fusion techniques. As these content based video recommendation models do not rely on user-video interaction behavior, they can be applied to new video recommendation without any historical behavior data [135], [136]. In contrast to the content-based recommendation models, with user-video interaction records, researchers proposed an Attentive Collaborative Filtering (ACF) model for multimedia recommendation [19]. ACF leverages the attention mechanism with visual inputs to learn the attentive weights to summarize users' preferences for historical items and the components of the item. Besides, researchers proposed a novel application of jointly recommending multimedia item and key frame recommendation without any user-frame interaction records. The key idea of this model is to leverage users' multimedia behavior and explicitly project users into two spaces: a collaborative space and a visual space, such that users' key frame preference could be approximated in the visual space. The authors designed a model to discern both the collaborative and visual dimensions of users, and model how users make decisive item preferences from these two aspects [133].

3.4 Modeling Social Network

With the emergence of social networks, users like to perform item preferences on these social platforms and share their interests with social connections. Social recommendation has emerged in these platforms, with the goal to model the social influence and social correlation among users to boost recommendation performance. The underlying reason for social recommendation is the existence of social influence among social neighbors, leading to the correlation of users' interests in a social network [145], [146], [147], [148], [149], [150], [151]. We summarize the social recommendation models into the following two categories: the social correlation enhancement and regularization models, and GNN based models.

Social Correlation Enhancement and Regularization. By treating users' social behavior as the social domain and item preference behavior as the item domain, the social correlation enhancement and regularization models tried to fuse users' two kinds of behaviors from two domains in a unified representation. For each user, her latent embedding p_u is composed of two parts: a free embedding e_u from the item domain, and a social embedding h_u that is similar with social connections in the social domain [145], [148], [150], [152], [153]. In other words, we have:

$$h_u = g(a; S) \tag{13}$$

$$p_u = f(e_u; h_a); \tag{14}$$

where g models the social embedding part with the social network structure as input, and f fuses the two kinds of embeddings, such as concatenation, addition or neural networks. Different models vary in the detailed implementation of the social domain representation h_u. E.g., it can be directly learned from the social network embedding models [145], aggregated from the social neighbors' embedding [148], [152], or transferred from the social domain to item domain with attention based transfer learning models [150]. Besides, the social network is also utilized as a regularization term in the model optimization process, with the assumption that connected users are more similar in the learned embedding space [145].

In the real-world, users' interests are dynamic over time due to users' personal interests change and the varying social influence strengths. Researchers extended the social correlation based model with RNN to model the evolution of users' preferences under dynamic social influences [146], [147]. Specifically, for each user u, her latent preferences h_a^t at time t could be modeled as the transition from her previous latent preference h_u^{t-1}, as well as the social influence from social neighbors at $t-1$ as:

$$h_u^t = f_{RNN}(R_u^t; h_u^{t-1}; \sum_{a \in S_u} t_{au} h_a^{t-1}) \tag{15}$$

where R_u^t is the temporal behaviors of user u at this time, $\sum_{a \in S_u} t_{au} h_a^{t-1}$ denotes the influences from her social neighbors. In particular, the social influence strength t_{au} could be simply set as equally for each social neighbor, or with attention modeling for influence strength inference.

GNN Based Approaches. Most of the above social recommendation models utilized the local first-order social neighbors for social recommendation. In the real world, the social diffusion process presents a dynamic recursive effect to influence a user's decision. In other words, each user is influenced recursively by the global social network graph structure. To this end, researchers argued that it is better to leverage the GNN based models to better model the global social diffusion process for recommendation. DiffNet is designed to simulate how users are influenced by the recursive social diffusion process for social recommendation with the social GNN modeling. Specifically, DiffNet recursively diffuses the social influence from step 0 to the stable diffusion depth K. Let h_u^k denote the user embedding at the k^{th} diffusion process, which is modeled as:

$$h_u^0 = f_{NN}(x_u; e_u) \tag{16}$$

$$h_{S_u}^{(k-1)} = Pool(h_a^{(k-1)} | a \in S_u) \tag{17}$$

$$h_u^k = \sigma(W^k[h_{S_u}^{k-1}; h_u^{(k-1)}]) \tag{18}$$

where Eq.(16) fuses the user feature x_u and user free latent vector e_u with a neural network f_{NN} for initial influence diffusion. At each diffusion step k, Eq.(17) models the influence diffusion from u's social neighbors, and Eq.(18) depicts the user embedding at the recursive step k by fusing her previous embedding h_u^{k-1} and influences from her social neighbors as $h_{S_u}^{k-1}$. As k diffuses from step 1 to depth K, the recursive social diffusion process is captured.

Instead of performing GNNs on the user-user social graph, researchers have also considered jointly modeling the social diffusion process in the social network and the interest diffusion process in the user-item graph with heterogeneous GNN based models [151], [154], [155], [156], [157], [158], [159]. E.g.,DiffNet++ is proposed to jointly model the interest diffusion from user-item bipartite graph and the influence diffusion from the user-user social graph for user modeling in social recommendation, and have achieved state-of-the-art performance [157].

3.5 Modeling Knowledge Graph

Researchers have also considered leveraging Knowledge Graphs (KG) for recommendation, which provide rich side information for items (*i.e.*, item attributes and external knowledge). Typically, it consists of real-world entities and relationships among them to profile an item. For example, a movie can be described by its director, cast, and genres. KG organizes such subject-property-object facts in the form of directed graph $G = \{(h, r, t) | h, t \in E, r \in R\}$, where each triplet presents that there is a relationship r from head entity h to tail entity t. Exploring such interlinks, as well as user-item interactions, being a promising solution to enrich item profile and enhance the relationships between users and items. Furthermore, such graph structure endows recommender systems the ability of reasoning and explainability [160], [161], [162], [163], [164], [165], [166]. Recent efforts for KG enhanced recommendation can be roughly categorized into three categories: path-based models [161], [167], [168], [169], [170], regularization-based models [125], [162], [171], [172], and GNN-based approaches [90], [121], [134], [163], [173], [174], [175], [176], [177], [178], [179].

Path Based Methods. Many efforts introduce meta-paths [167], [169], [170], [180], [181], [182], [183], [184] and paths [161], [168], [185], [186], [187] that present high-order connectivity between users and items, and then feed them into predictive models to directly infer user preferences. In particular, a path from user u to item i can be defined as a sequence of entities and relations: $p = [e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} \cdots \xrightarrow{r_{L-1}} e_L]$, where $e_1 = u$ and $e_L = i$, and (e_l, r_l, e_{l+1}) is the l-th triplet in p, and $L-1$ denotes the number of triplets in the path. As such, the set of paths connecting u and i can be defined as $P(u, i) = \{p\}$. To handle the extreme large number of paths, either path selection algorithm to select prominent paths or defined meta-path patterns to constrain the paths are applied.

FMG [169], MCRec [170], and KPRN [161] convert the path set into an embedding vector to represent the user-item connectivity. Such paradigm can be summarized as follows:

$$c = f_{Pooling}(\{f_{Embed}(p) | p \in P(u, i)\}), \tag{19}$$

where $f_{Embed}(\cdot)$ embeds path p as a trainable vector. $f_{Pooling}(\cdot)$ is the pooling operation to synthesize all path information into the connectivity representation, such as the attention networks adopted in MCRec and KPRN. RippleNet [188] constructs ripple set (*i.e.*, high-order neighboring items derived from P) for each user to enrich her representations.

While explicitly modeling high-order connectivity, it is highly challenging in real-world recommendation scenarios because most of these methods require extensive domain knowledge to define meta-paths or labor-intensive feature engineering to obtain qualified paths [160], [163]. Moreover, the scale of paths can easily reach millions or even larger when a large number of KG entities are involved, making it prohibitive to efficiently transfer knowledge.

Regularization Based Methods. This research line devises a joint learning framework, where direct user-item interactions are used to optimize the recommender loss, and KG triples are utilized as additional loss terms to regularize the recommender model learning.In particular, the anchors between two modeling components are the embeddings of the overlapped items. CKE [125] makes use of Knowledge Graph Embedding (KGE) techniques, especially TransR [189], to generate additional representations of items, and then integrates them with item embeddings of the recommender MF, which is defined as:

$$q_i = f_{Embed}(i) + f_{KGE}(i|G), \tag{20}$$

where $f_{Embed}(\cdot)$ is the embedding function which takes the item ID as the input, while f_{KGE} is the output of KGE method which considers the KG structure. Similarly, DKN [190] generates item embeddings from NCF and TransE. These approaches focus on enriching item representations by the joint learning framework. KTUP [162] and CFKG [191] reorganize user-item interactions in the form of triples by involving a new interaction relation, so as to be seamlessly integrated with KG triplets, casting the recommendation task as the link prediction, or more precisely, KG completion task.

GNN Based Methods. The regularization-based methods only take direct connectivity between entities into consideration, while encoding the high-order connectivity in a rather implicit manner. Due to the lack of explicit modeling, neither the long-range connectivities are guaranteed to be captured, nor the results of high-order modeling are interpretable [163]. More recent works, such as KGAT [163], CKAN [192], MKM-SR [193], and KGCN [173], get inspired by the advances of GNNs and explore the message-passing mechanism over graphs to exploit high-order connectivity in an end-to-end fashion.

KGAT [163] encodes user-item interactions and KG as a unified relational graph by representing each user behavior as a triplet, $(u, \text{Interact}, i)$. Based on the item-entity alignment set, the user-item bipartite graph can be seamlessly integrated with KG as a so-called collaborative knowledge graph $G = f(h;r;t)jh;t\ 2\ E^0;r\ 2\ Rg$, where $E^0 = E\ [\ U$ and $R^0 = R\ [\ f\ \text{Interact}g$. Over such graph, KGAT recursively propagates the embeddings from a node's neighbors (which can be users, items, or other entities) to refine the node's embedding, and employs an attention mechanism to discriminate the importance of the neighbors as:

$$p_u = f_{GNN}(u;G); \tag{21}$$

where $f_{GNN}(\ )$ is the GNN component.

4 Temporal/Sequential Models

Users' preferences are not static but evolve over time. Instead of modeling users' static preferences with the aforementioned models, temporal/sequential based recommendation focuses on modeling users' dynamic preferences or sequential patterns over time. Given a userset $U = [u_1; u_2; :::; u_M]$ and an itemset $V = [i_1; i_2; :::; i_N]$, current temporal/sequential recommendation could be generally classified into three categories:

- Temporal based recommendation: For a user $u\ 2\ U$ and an item $i\ 2\ V$, the associated user-item interaction behavior is denoted as a quadri-tuple as $[u; i; r_{ui}; t]$. In this representation, r_{ui} denotes the detailed rating and t_{ui} is the timestamp of this behavior. Temporal recommendation focuses on modeling the temporal dynamics of users' behavior over time.
- Session based recommendation: In a certain session $s = [i_1; i_2; :::; i_{jsj}]$ $(s\ \ V)$, a user interacts with a collection of items (e.g., consumption with a shopping basket, browsing the internet in a limited time period). In many session based applications, users do not log in and user IDs are not available [194], [195], [196]. Therefore, the popular direction of session based recommendation is to mine the sequential item-item interaction patterns from the session data for better recommendation.
- Temporal and session based recommendation: This approach combines the definition of temporal recommendation and session recommendation, in which each transaction is described as $[u; s; t]$, with $s\ \ V$ is a collection of items that are consumed at a particular time t. Under this scenario, both the temporal evolution and the sequential patterns of items need to be captured.

We summarize the main techniques for modeling the temporal and sequential effects in recommender systems in Table 5.

4.1 Temporal based recommendation

Temporal recommendation models focus on capturing the temporal evolution of users' preferences over time. Due to the superior performance of RNNs in modeling temporal patterns, many temporal based recommendation approaches model the evolution of users' latent vectors or item latent vectors with RNNs. Recurrent Recommender Networks (RRN) is one of the representative works for temporal recommendation by endowing both users and items with an LSTM autoregressive architecture [197]. In RRN, the predicted rating $\hat{r}^t_{ui}$ of user u to item i at time t is modeled as:

$$\hat{r}^t_{ui} = f(p^t_u; q^t_i) \quad \text{where} \tag{22}$$

$$p^t_u = RNN(p^{(t\ \ 1)}_u; Wx^t_u); \quad q^t_i = RNN(q^{(t\ \ 1)}_i; Wx^t_i) \tag{23}$$

where p^t_u and q^t_i are the dynamic embeddings of user u and item i at time t, respectively. Specifically, f in Eq.(22) is a temporal rating prediction function with the user and item dynamic embeddings at that time. Eq.(23) models the evolution of users and items' dynamic embeddings with RNN architecture. As the user side and item side share similar LSTM structure, we take the user side as an example. $x^t_u\ 2\ R^{jVj}$ is a rating vector for u between $t\ \ 1$ and current time t, with each element x^t_{ul} denotes the rating of user u to each item l at that time. W is a transformation matrix that needs to be learned. Therefore, RRN learns the evolution of user and item latent vectors over time with two RNNs. Many temporal recommendation models extended the basic ideas of RRN by considering rich context factors, such as the social influence [146], [200], item metadata [201], [242] and multimedia data fusion [243]. Take the RNN in the user side as an example, and we can generalize the user latent embedding evolution as:

$$p^t_u = RNN(p^{(t\ \ 1)}_u; Wx^t_u; ContextualEmbedding); \tag{24}$$

where the additional contextual embeddings are also injected to model the temporal evolution of users' temporal embedding.

Recently, an emerging trend is to model the temporal evolution with Neural Turning Machines [244] and Memory Networks [245]. Compared to RNNs, memory networks introduce a memory matrix to store the states in memory slots, and update memories over time with read and write operations. As the memory storage is limited, the key component in applying memory networks in recommendation is how to update memories over time with users' temporal behavior. Researchers proposed a general memory augmented neural network with user memory networks to store and update users' historical records, and the user memory network is implemented from the item and feature level [204]. Researchers further proposed to use attention mechanism in the memory reading and writing process with soft-addressing, in order to better capture users' long-term stable and short-term temporal interests [203].

4.2 Session based recommendation

Many real-world recommender systems often encounter the short session data from anonymous users, i.e., the user ID information is not available. Session based recommendation

TABLE 5: A comparison of methods that models the temporal and sequential effects in RS

Model Type	Model Summarization	Models
Temporal Models	Recurrent neural networks to capture temporal evolution	ARSE [146],RRN [197] [198], [199], [200], [201], [202]
	Memory network based models	NMRN [203], MANN [204]
Sequential Models	RNN based models that rely on sessions to construct input and output	p-RNN [195],KERL [185],CRNNs [205], NARM [206], [207], [208], [209], [194],
	Translation based models for modeling the correlations of consecutive items	TransRec [39],PeterRec [210], [211]
	Convolutional sequence embedding models	3D CNNs [212]
	Self attention for learning item correlations	SASRec [213], MFGAN [214]
	Memory network to learn the session representation	DMN [215]
	GNN based models for learning item correlations	SR-GNN [196],GC-SAN [216],Gag [217], GCE-GNN [218],SGNN-HN [219], [220]
Temporal and Sequential Models	Hierarchical attention networks with long and short term interest	SHAN [221],HRM [222],HGN [223], MARank [224],Fissa [225],SSE-PT [226]
	RNN based models	RRN [227],BINN [197],HIERNN [228]
	Attention based models for user interest modeling	CTRec [229],M3 [230],S3-rec [231],CTA [232], MTAM [233],TASER [234],ReChorus [235]
	Memory Networks for long distance item correlations	KA-MemNN [236],CSRM [237],MTAM [233]
	CNN based models	Caser [43],CTRec [229], [238]
	GNN based models	HyperRec [239],MA-GNN [240],IMfOU [241]

is popular under this situation, which models the sequential item transition patterns given many session records. Hidasi et al. made one of the first few attempts to design GRU4REC for session based recommendation under the RNN based framework [194], [246], [247]. Specifically, GRU4REC resembles an RNN structure, which recursively takes the current item in the session as input, updates the hidden states, and outputs the predicted next item based on the hidden state. Given anonymous sessions, the key component of GRU4REC is how to construct mini-batches to suit the data forms of RNNs. Since the goal is to capture how a session evolves over time with item dependencies, the authors designed a session parallel min-batches. The first event of the first several sessions are extracted to form the first mini-batch, with the desired output is the second event of the corresponding session. Under such a formulation, the complex correlations of items in a session are captured for session based recommendation.

GRU4REC has been further investigated with item feature consideration [195], local intent [206], user information consideration [248], data augmentation techniques [207]. By treating item ID, name, and category with an embedding matrix that ensembles a frame, a sequence of clicks could be represented as frames. Therefore, the architecture of 3D CNNs could be transferred to session-based recommendation [212]. Furthermore, a self attention based sequential model of SASRec is proposed. SASRec models the entire user sequence without any recurrent and convolutional operations, and adaptively considers consumed items for recommendation [213].

Researchers also proposed a translation based model to capture the personalized sequential third order interactions between a user u, the previous item j, and the current item j. Given the item embedding matrix Q, each user's embedding p_u can be approximated as: $q_i + p_u \approx q_j$ [39]. Therefore, the translation based models capture the correlation of two constructive items.

While the above models built the relationships between consecutive items in a session, how to globally model the transitions in a session among distant items remain under explored. Researchers adopted GNNs for session based recommendation [196], [216], [217], [218], [219], [220]. SR-GNN is one of the first few attempts for GNN based session recommendation. In this figure, the graph is constructed by taking all items as the graph node set, and there is an edge between two nodes if these two nodes appear in consecutive orders in a session. Then, the GNN is adopted to learn item embeddings, such that the higher-order relationships of items from session behavior data can be modeled [196]. Different GNN based models vary in graph construction, and graph aggregation process [216], [217], [218].

4.3 Temporal and session based recommendation

Given the session data of each user over time, models in this category leverage both the temporal evolution modeling of users, as well as the sequential item patterns hidden in the sessions for recommendation. Currently, the solutions to this problem could be classified into two categories: the first category learns both users' long term preference and the short term dynamic preferences, and the second category adopts advanced neural models for learning a unified user representation.

In the first category, each user's long term preference is modeled from her historical behaviors, and the short term dynamics is modeled from the previous session or the current session [221], [222], [224] . For example, researchers proposed hierarchical attention networks for temporal and session based recommendation, with the first attention layer learns the user long term preference based on historical records, and the second one attentively aggregates user representation from the current session as:

$$p^t_u = Att_2(p_u; Att_1(q_l; l \in T^{(t-1)}_u)); \qquad (25)$$

where Att_1 denotes the bottom layer attention network that depicts the user's short term preference from the recent user behavior $T^{(t-1)}_u$, and Att_2 is a top layer attention network that attentively balances the short term user preference and her long term preference embedding vector p_u. Instead of using hierarchical attentions, researchers proposed to adopt attention techniques to learn item correlations, and designed recurrent states at top layers for sequential recommendation [232].

Researchers also proposed hierarchical RNNs for personalized session-based recommendation over time, with a session level GRU unit to model the user activity within sessions, and a user level GRU models the evolution of the user preference over time [193], [228]. Besides, researchers exploited hierarchical attention networks to learn better short term user preference with feature-level attention and item level attention [223]. For the long time user interest modeling, researchers proposed to leverage nearby sessions [227], designed attention modeling or memory addressing techniques to find related sessions [229], [236], [237], [249].

Another kind of models utilize the 3D convolutional networks for recommendation, which defines the recommendation problem as [43], [238]: $(S^u_{t-L}, \ldots, S^u_{t-2}, S^u_{t-1}) \rightarrow S^u_t$, where $S^u_t \subseteq V$ is the t-th time sequential behavior of user u at time t, and L denotes the maximum sequence length. Convolutional Sequence Embedding Recommendation (Caser) is a representative work that incorporates CNNs to learn the sequential patterns. It captures both user's general preferences and sequential patterns, at both the union level and point level with convolution operations, and captures the skip behavior [43], [238].

Besides, researchers proposed to leverage the advances of GNN based models for recommendation [239], [240], [241]. The graph structure is constructed from all sessions to form a global item correlation graph or graphs at each time period. E.g., researchers constructed time-aware hypergraphs to model item correlations over time. After that, the self attention modules are used to model users' dynamic interests based on the learned dynamic item embeddings over time [239].

5 Conclusion and Future Directions

We hope this survey will give researchers a comprehensive picture of state-of-the-art models for neural network based recommendation. The foregoing various neural network based recommendation models have demonstrated the superior recommendation quality. Meanwhile, we realize that current solutions for recommendation are far from satisfactory, and there are still much opportunities in this area. We discuss some possible directions that deserve more research efforts from the basics, modeling and applications perspectives.

Basics: Recommendation Benchmarking. While the field of neural recommender systems has seen a great surge of interests in recent years, it has also been difficult for researchers to keep track of what represents the state-of-the-art models [250]. It is urgent to identify the architectures and key mechanisms that generalize to most recommender models. However, this is a non-trivial task as recommendation scenarios are diverse, e.g., static recommendation models or dynamic recommendation models, content enriched or knowledge enhanced models. Different recommendation models rely on different data sets with varying inputs. Besides, the same model would have varying performance on different recommendation scenarios due to the assumption in the modeling process. In fact, the Netflix competition for CF based recommendation has passed more than 10 years, how to design a large benchmarking recommendation dataset that keeps track of the state-of-the-art recommendation problems and update the leading performance for comparisons is a challenging yet urgent future direction.

Models: Graph Reasoning based Recommendation. Graphs are ubiquitous structures in representing various recommendation scenarios. E.g., CF could be seen as a user-item bipartite graph, content based recommendation is represented as an attributed user-item bipartite graph or a heterogeneous information network [144], [251], and knowledge enhanced recommendation is defined as a combination of knowledge graph and user-item bipartite graph. With the great success of deep learning on graphs [144], it is promising to design graph based models for recommendation. Some recent works have empirically demonstrated the superiority of graph embedding based recommendation models, how to explore the natural graph reasoning techniques for better recommendation is a promising direction.

HCI: Conversational Recommendation. Most current recommender sysetms present in a single round form of human-computer interaction (HCI). These systems suffer from limited data for precisely inferring users' preferences. In the real-world, human communicate with others in multiple rounds to make decisions. With the recent progress of natural language dialogs between human and AI agents (a.k.a., chatbots), building conversational AI agents for recommender sysetms that present multiple round interactions between users and AI systems is a promising direction [252]. Conversational recommender systems need to model user interests, design conversational strategies, formulate natural language, and consider exploit-explore balance in the dialogue process [109], [253], [254].

Evaluation: Multi-Objective Goals for Social Good Recommendation. Recommender systems have penetrated every aspect in our daily life, and have greatly shaped the decision process of providers and users. Most previous recommender systems concentrated on the single goal of recommendation accuracy based user experience. These systems limit the ability to incorporate user satisfaction from multiple goals, e.g., recommendation diversity and explanations to persuade users [9]. Besides, the user-centric approach neglects system objectives from multistakeholders and the society. The data-driven approaches with accuracy as goals may lead to biases in the algorithmic process decision process [255], [256], [257]. For recommender systems, researchers have realized that long tailed items have fewer chances to be recommended, and benefiting users may obscure concerns that might come from other stakeholders in this system. How to provide multi-objective goals for social good recommendation, such as explainability, balance of multistakeholders, and fairness for the society is an important research topic that needs to be paid attention to.

References

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,”

Commun. ACM, vol. 35, no. 12, pp. 61–70, 1992.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE TKDE, vol. 17, no. 6, pp. 734–749, 2005.

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, no. 8, pp. 30–37, 2009.

[4] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in RecSys, 2010, pp. 79–86.

[5] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in WWW, 2010, pp. 811–820.

[6] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep learning. MIT press Cambridge, 2016, vol. 1, no. 2.

[7] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," ACM CSUR, vol. 47, no. 1, pp. 1–45, 2014.

[8] M. M. Khan, R. Ibrahim, and I. Ghani, "Cross domain recommender systems: A systematic literature review," ACM CSUR, vol. 50, no. 3, pp. 1–34, 2017.

[9] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," TRIR, vol. 14, no. 1, pp. 1–101, 2020.

[10] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He, "A survey on knowledge graph-based recommender systems," IEEE TKDE, 2020, preprint.

[11] H. Wang, X. Shi, and D.-Y. Yeung, "Relational stacked denoising autoencoder for tag recommendation," in AAAI, 2015, pp. 3052–3058.

[12] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in UAI, 2009, pp. 452–461.

[13] S. Kabbur, X. Ning, and G. Karypis, "FISM: factored item similarity models for top-n recommender systems," in SIGKDD, 2013, pp. 659–667.

[14] C. Ma, L. Ma, Y. Zhang, R. Tang, X. Liu, and M. Coates, "Probabilistic metric learning with adaptive margin for top-k recommendation," in SIGKDD, 2020, pp. 1036–1044.

[15] D. Lian, X. Xie, E. Chen, and H. Xiong, "Product quantized collaborative filtering," IEEE TKDE, 2020, preprint.

[16] J. Chen, C. Wang, S. Zhou, Q. Shi, J. Chen, Y. Feng, and C. Chen, "Fast adaptively weighted matrix factorization for recommendation with implicit feedback," in AAAI, 2020, pp. 3470–3477.

[17] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in SIGKDD, 2008, pp. 426–434.

[18] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "Nais: Neural attentive item similarity model for recommendation," IEEE TKDE, vol. 30, no. 12, pp. 2354–2366, 2018.

[19] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in SIGIR, 2017, pp. 335–344.

[20] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in WWW, 2015, pp. 111–112.

[21] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in WSDM, 2016, pp. 153–162.

[22] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in WWW, 2018, pp. 689–698.

[23] F. Zhuang, D. Luo, N. J. Yuan, X. Xie, and Q. He, "Representation learning with pair-wise constraints for collaborative ranking," in WSDM, 2017, pp. 567–575.

[24] K. Luo, H. Yang, G. Wu, and S. Sanner, "Deep critiquing for vae-based recommender systems," in SIGIR, 2020, pp. 1269–1278.

[25] F. Khawar, L. Poon, and N. L. Zhang, "Learning the structure of auto-encoding recommenders," in WWW, 2020, pp. 519–529.

[26] R. van den Berg, T. Kipf, and M. Welling, "Graph convolutional matrix completion," ArXiv, vol. abs/1706.02263, 2017.

[27] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in SIGIR, 2019, pp. 165–174.

[28] L. Zheng, C. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," in RecSys, 2018, pp. 311–319.

[29] J. Sun, Y. Zhang, W. Guo, H. Guo, R. Tang, X. He, C. Ma, and M. Coates, "Neighbor interaction aware graph convolution networks for recommendation," in SIGIR, 2020, pp. 1289–1298.

[30] J. Sun, W. Guo, D. Zhang, Y. Zhang, F. Regol, Y. Hu, H. Guo, R. Tang, H. Yuan, X. He et al., "A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks," in SIGKDD, 2020, pp. 2030–2039.

[31] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua, "Disentangled graph collaborative filtering," in SIGIR, 2020, pp. 1001–1010.

[32] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in AAAI, 2020, pp. 27–34.

[33] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in SIGIR, 2020, pp. 639–648.

[34] S. Ji, Y. Feng, R. Ji, X. Zhao, W. Tang, and Y. Gao, "Dual channel hypergraph collaborative filtering," in SIGKDD, 2020, pp. 2020–2029.

[35] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," ACM TOIS, vol. 37, no. 3, pp. 1–25, 2019.

[36] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in SIGKDD, 2018, pp. 1059–1068.

[37] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in ICLR, 2017.

[38] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," in WWW, 2017, pp. 193–201.

[39] R. He, W.-C. Kang, and J. McAuley, "Translation-based recommendation," in RecSys, 2017, pp. 161–169.

[40] Y. Tay, L. Anh Tuan, and S. C. Hui, "Latent relational metric learning via memory-based attention for collaborative ranking," in WWW, 2018, pp. 729–739.

[41] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in WWW, 2017, pp. 173–182.

[42] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T. Chua, "Outer product-based neural collaborative filtering," in IJCAI, 2018, pp. 2227–2233.

[43] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in WSDM, 2018, pp. 565–573.

[44] F. Strub and J. Mary, "Collaborative filtering with stacked denoising autoencoders and sparse inputs," in NeurIPS workshop, 2015.

[45] S. Zhang, L. Yao, and X. Xu, "Autosvd++: An efficient hybrid collaborative filtering model via contractive auto-encoders," in SIGIR, 2017, pp. 957–960.

[46] F. Zhuang, Z. Zhang, M. Qian, C. Shi, X. Xie, and Q. He, "Representation learning via dual-autoencoder for recommendation," Neural Networks, vol. 90, pp. 83–89, 2017.

[47] S. Rendle, "Factorization machines," in ICDM, 2010, pp. 995–1000.

[48] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for ctr prediction," in RecSys, 2016, pp. 43–50.

[49] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in SIGIR, 2017, pp. 355–364.

[50] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data," in ECIR, 2016, pp. 45–57.

[51] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in ICDM, 2016, pp. 1149–1154.

[52] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. Mao, "Deep crossing: Web-scale modeling without manually crafted combinatorial features," in SIGKDD, 2016, pp. 255–262.

[53] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in RecSys, 2016, pp. 191–198.

[54] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir et al., "Wide & deep learning for recommender systems," in DLRS, 2016, pp. 7–10.

[55] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for CTR prediction," in IJCAI, 2017, pp. 1725–1731.

[56] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in ADKDD, 2017, pp. 1–7.

[57] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, “xdeepfm: Combining explicit and implicit feature interactions for recommender systems,” in SIGKDD, 2018, pp. 1754–1763.

[58] X. Wang, X. He, F. Feng, L. Nie, and T.-S. Chua, “Tem: Tree-enhanced embedding model for explainable recommendation,” in WWW, 2018, pp. 1543–1552.

[59] M. Blondel, A. Fujino, N. Ueda, and M. Ishihata, “Higher-order factorization machines,” in NeurIPS, 2016, pp. 3351–3359.

[60] D. W. Otter, J. R. Medina, and J. K. Kalita, “A survey of the usages of deep learning for natural language processing,” IEEE TNNLS, 2020, preprint.

[61] S. Lappin, Deep Learning and Linguistic Representation. CRC Press, 2021.

[62] L. Zheng, V. Noroozi, and P. S. Yu, “Joint deep modeling of users and items using reviews for recommendation,” in WSDM, 2017, pp. 425–434.

[63] P. Sun, L. Wu, K. Zhang, Y. Fu, R. Hong, and M. Wang, “Dual learning for explainable recommendation: Towards unifying user preference prediction and review generation,” in WWW, 2020, pp. 837–847.

[64] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative deep learning for recommender systems,” in SIGKDD, 2015, pp. 1235–1244.

[65] H. Wang, S. Xingjian, and D.-Y. Yeung, “Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks,” in NeurIPS, 2016, pp. 415–423.

[66] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, “A hybrid collaborative filtering model with deep structure for recommender systems.” in AAAI, 2017, pp. 1309–1315.

[67] X. Li and J. She, “Collaborative variational autoencoder for recommender systems,” in SIGKDD, 2017, pp. 305–314.

[68] H. Ying, L. Chen, Y. Xiong, and J. Wu, “Collaborative deep ranking: A hybrid pair-wise recommendation algorithm with implicit feedback,” in PAKDD, 2016, pp. 555–567.

[69] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, “Collaborative filtering and deep learning based recommendation system for cold start items,” Expert Systems with Applications, vol. 69, pp. 29–39, 2017.

[70] Z. Xu, T. Lukasiewicz, C. Chen, Y. Miao, and X. Meng, “Tag-aware personalized recommendation using a hybrid deep model,” in IJCAI, 2017, pp. 3196–3202.

[71] F. Strub, R. Gaudel, and J. Mary, “Hybrid recommender system based on autoencoders,” in DLRS, 2016, pp. 11–16.

[72] C. Ma, P. Kang, B. Wu, Q. Wang, and X. Liu, “Gated attentive-autoencoder for content-aware recommendation,” in WSDM, 2019, pp. 519–527.

[73] S. Okura, Y. Tagami, S. Ono, and A. Tajima, “Embedding-based news recommendation for millions of users,” in SIGKDD, 2017, pp. 1933–1942.

[74] W. Fu, Z. Peng, S. Wang, Y. Xu, and J. Li, “Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems,” in AAAI, 2019, pp. 94–101.

[75] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, “Convolutional matrix factorization for document context-aware recommendation,” in RecSys, 2016, pp. 233–240.

[76] R. Catherine and W. Cohen, “Transnets: Learning to transform for recommendation,” in RecSys, 2017, pp. 288–296.

[77] T. Ebesu and Y. Fang, “Neural citation network for context-aware citation recommendation,” in SIGIR, 2017, pp. 1093–1096.

[78] H. Lee, Y. Ahn, H. Lee, S. Ha, and S.-g. Lee, “Quote recommendation in dialogue using deep neural network,” in SIGIR, 2016, pp. 957–960.

[79] C.-Y. Wu, A. Ahmed, A. Beutel, and A. J. Smola, “Joint training of ratings and reviews with recurrent recommender networks,” in ICLR Workshop, 2017.

[80] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, and Y. Li, “Metapath-guided heterogeneous graph neural network for intent recommendation,” in SIGKDD, 2019, pp. 2478–2486.

[81] D. Liu, J. Lian, S. Wang, Y. Qiao, J.-H. Chen, G. Sun, and X. Xie, “Kred: Knowledge-aware document representation for news recommendations,” in RecSys, 2020, pp. 200–209.

[82] Y. Kim, “Convolutional neural networks for sentence classification,” in EMNLP, 2014, pp. 1746–1751.

[83] T. Bansal, D. Belanger, and A. McCallum, “Ask the gru: Multi-task learning for deep text recommendations,” in RecSys, 2016, pp. 107–114.

[84] Y. Gong and Q. Zhang, “Hashtag recommendation using attention-based convolutional neural network,” in IJCAI, 2016, pp. 2782–2788.

[85] S. Seo, J. Huang, H. Yang, and Y. Liu, “Interpretable convolutional neural networks with dual local and global attention for review rating prediction,” in RecSys, 2017, pp. 297–305.

[86] Y. Li, T. Liu, J. Jiang, and L. Zhang, “Hashtag recommendation with topical attention-based LSTM,” in COLING, 2016, pp. 3019–3029.

[87] C. Qin, H. Zhu, C. Zhu, T. Xu, F. Zhuang, C. Ma, J. Zhang, and H. Xiong, “Duerquiz: A personalized question recommender system for intelligent job interview,” in SIGKDD, 2019, pp. 2165–2173.

[88] H. Wang, F. Wu, Z. Liu, and X. Xie, “Fine-grained interest matching for neural news recommendation,” in ACL, 2020, pp. 836–845.

[89] T. Qi, F. Wu, C. Wu, Y. Huang, and X. Xie, “Privacy-preserving news recommendation model learning,” in EMNLP, 2020, pp. 1423–1432.

[90] D. Lee, B. Oh, S. Seo, and K.-H. Lee, “News recommendation with topic-enriched knowledge graphs,” in CIKM, 2020, pp. 695–704.

[91] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. S. Kankanhalli, “Aˆ3ncf: An adaptive aspect attention model for rating prediction,” in IJCAI, 2018, pp. 3748–3754.

[92] Q. Zhu, X. Zhou, Z. Song, J. Tan, and L. Guo, “Dan: Deep attention neural network for news recommendation,” in AAAI, 2019, pp. 5973–5980.

[93] X. Chen, Y. Zhang, and Z. Qin, “Dynamic explainable recommendation based on neural attentive models,” in AAAI, 2019, pp. 53–60.

[94] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, and X. Xie, “Npa: Neural news recommendation with personalized attention,” in SIGKDD, 2019, pp. 2576–2584.

[95] Y. Lu, R. Dong, and B. Smyth, “Coevolutionary recommendation model: Mutual learning between ratings and reviews,” in WWW, 2018, pp. 773–782.

[96] D. Liu, J. Li, B. Du, J. Chang, and R. Gao, “Daml: Dual attention mutual learning between ratings and reviews for item recommendation,” in SIGKDD, 2019, pp. 344–352.

[97] L. Hu, S. Jian, L. Cao, and Q. Chen, “Interpretable recommendation via attraction modeling: Learning multilevel attractiveness over multimodal movie contents.” in IJCAI, 2018, pp. 3400–3406.

[98] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, and X. Xie, “Neural news recommendation with attentive multi-view learning,” in IJCAI, 2019, pp. 3863–3869.

[99] J. Gao, Y. Lin, Y. Wang, X. Wang, Z. Yang, Y. He, and X. Chu, “Set-sequence-graph: A multi-view approach towards exploiting reviews for recommendation,” in CIKM, 2020, pp. 395–404.

[100] R. Ma, Q. Zhang, J. Wang, L. Cui, and X. Huang, “Mention recommendation for multimodal microblog with cross-attention memory network,” in SIGIR, 2018, pp. 195–204.

[101] Q. Zhang, J. Wang, H. Huang, X. Huang, and Y. Gong, “Hashtag recommendation for multimodal microblog using co-attention network,” in IJCAI, 2017, pp. 3420–3426.

[102] C. Chen, M. Zhang, Y. Liu, and S. Ma, “Neural attentional rating regression with review-level explanations,” in WWW, 2018, pp. 1583–1592.

[103] X. Wang, Y. Chen, J. Yang, L. Wu, Z. Wu, and X. Xie, “A reinforcement learning framework for explainable recommendation,” in ICDM, 2018, pp. 587–596.

[104] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in ICLR, 2015.

[105] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam, “Neural rating regression with abstractive tips generation for recommendation,” in SIGIR, 2017, pp. 345–354.

[106] P. Li, Z. Wang, L. Bing, and W. Lam, “Persona-aware tips generation,” in WWW, 2019, pp. 1006–1016.

[107] Z. Chen, X. Wang, X. Xie, T. Wu, G. Bu, Y. Wang, and E. Chen, “Co-attentive multi-task learning for explainable recommendation,” in IJCAI, 2019, pp. 2137–2143.

[108] L. Li, Y. Zhang, and L. Chen, “Generate neural template explanations for recommendation,” in CIKM, 2020, pp. 755–764.

[109] Z. Chen, X. Wang, X. Xie, M. Parsana, A. Soni, X. Ao, and E. Chen, “Towards explainable conversational recommendation,” in IJCAI, 2020, pp. 2994–3000.

[110] X. Chen, H. Chen, H. Xu, Y. Zhang, Y. Cao, Z. Qin, and H. Zha, "Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation," in SIGIR, 2019, pp. 765–774.

[111] Q.-T. Truong and H. Lauw, "Multimodal review generation for recommender systems," in WWW, 2019, pp. 1864–1874.

[112] R. He and J. McAuley, "VBPR: visual bayesian personalized ranking from implicit feedback," in AAAI , 2016, pp. 144–150.

[113] Y. Lin, M. Moosaei, and H. Yang, "Outfitnet: Fashion outfit recommendation with attention-based multiple instance learning," in WWW, 2020, pp. 77–87.

[114] W. Yu, H. Zhang, X. He, X. Chen, L. Xiong, and Z. Qin, "Aesthetic-based clothing recommendation," in WWW, 2018, pp. 649–658.

[115] J. Wen, X. Li, J. She, S. Park, and M. Cheung, "Visual background recommendation for dance performances using dancer-shared images," in iThings, 2016, pp. 521–527.

[116] M. Hou, L. Wu, E. Chen, Z. Li, Z. Vincent W., and Q. Liu, "Explainable fashion recommendation: A semantic attribute region guided approach," in IJCAI, 2019, pp. 4681–4688.

[117] T. Alashkar, S. Jiang, S. Wang, and Y. Fu, "Examples-rules guided deep neural network for makeup recommendation." in AAAI , 2017, pp. 941–947.

[118] X. Yang, X. He, X. Wang, Y. Ma, F. Feng, M. Wang, and T.-S. Chua, "Interpretable fashion matching with rich attributes," in SIGIR, 2019, pp. 775–784.

[119] S. Liu, Z. Chen, H. Liu, and X. Hu, "User-video co-attention network for personalized micro-video recommendation," in WWW, 2019, pp. 3020–3026.

[120] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in SIGKDD, 2018, pp. 974–983.

[121] X. Li, X. Wang, X. He, L. Chen, J. Xiao, and T.-S. Chua, "Hierarchical fashion graph network for personalized outfit recommendation," in SIGIR, 2020, pp. 159–168.

[122] L. Wu, Y. Yang, L. Chen, D. Lian, R. Hong, and M. Wang, "Learning to transfer graph embeddings for inductive graph based recommendation," in SIGIR, 2020, pp. 1211–1220.

[123] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in WWW, 2016, pp. 507–517.

[124] Q. Xu, F. Shen, L. Liu, and H. T. Shen, "Graphcar: Content-aware multimedia recommendation with graph autoencoder," in SIGIR, 2018, pp. 981–984.

[125] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in SIGKDD, 2016, pp. 353–362.

[126] X. Yang, Y. Ma, L. Liao, M. Wang, and T.-S. Chua, "Transnfcm: Translation-based neural fashion compatibility modeling," in AAAI , 2019, pp. 403–410.

[127] Â. Cardoso, F. Daolio, and S. Vargas, "Product characterisation towards personalisation: Learning attributes from unstructured data to recommend fashion products," in SIGKDD, 2018, pp. 80–89.

[128] X. Chen, Y. Zhang, Q. Ai, H. Xu, J. Yan, and Z. Qin, "Personalized key frame recommendation," in SIGIR, 2017, pp. 315–324.

[129] T. Yu, Y. Shen, and H. Jin, "Towards hands-free visual dialog interactive recommendation," in AAAI , 2020, pp. 1137–1144.

[130] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in MM , 2014, pp. 627–636.

[131] A. v. d. Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in NeurIPS, 2013, pp. 2643–2651.

[132] Z. Nazari, C. Charbuillet, J. Pages, M. Laurent, D. Charrier, B. Vecchione, and B. Carterette, "Recommending podcasts for cold-start users based on music listening and taste," in SIGIR, 2020, pp. 1041–1050.

[133] L. Wu, L. Chen, Y. H. Yang, R. Hong, Y. Ge, X. Xie, and M. Wang, "Personalized multimedia item and key frame recommendation," in IJCAI, 2019, pp. 1431–1437.

[134] L. Wu, Y. Yang, K. Zhang, R. Hong, Y. Fu, and M. Wang, "Joint item recommendation and attribute inference: An adaptive graph convolutional network approach," in SIGIR, 2020, pp. 679–688.

[135] J. Lee, S. Abu-El-Haija, B. Varadarajan, and A. Natsev, "Collaborative deep metric learning for video understanding," in SIGKDD, 2018, pp. 481–490.

[136] J. Lee and S. Abu-El-Haija, "Large-scale content-only video recommendation," in ICCVW, 2017, pp. 987–995.

[137] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in SIGIR, 2015, pp. 43–52.

[138] Y. S. Rawat and M. S. Kankanhalli, "Contagnet: Exploiting user context for image tag recommendation," in MM , 2016, pp. 1102–1106.

[139] C. Lei, D. Liu, W. Li, Z.-J. Zha, and H. Li, "Comparative deep learning of hybrid representations for image recommendations," in CVPR, 2016, pp. 2545–2553.

[140] Q. Liu, S. Wu, and L. Wang, "Deepstyle: Learning user preferences for visual recommendation," in SIGIR, 2017, pp. 841–844.

[141] W. Niu, J. Caverlee, and H. Lu, "Neural personalized ranking for image recommendation," in WSDM, 2018, pp. 423–431.

[142] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu, "What your images reveal: Exploiting visual contents for point-of-interest recommendation," in WWW, 2017, pp. 391–400.

[143] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in ICLR, 2017.

[144] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," IEEE TKDE, 2020, preprint.

[145] L. Wu, P. Sun, R. Hong, Y. Ge, and M. Wang, "Collaborative neural social recommendation," IEEE TSMC:S, vol. 51, no. 1, pp. 464–476, 2021.

[146] P. Sun, L. Wu, and M. Wang, "Attentive recurrent social recommendation," in SIGIR, 2018, pp. 185–194.

[147] P. Wu, Y. Tu, X. Yuan, A. Jatowt, and Z. Yang, "Neural framework for joint evolution modeling of user feedback and social links in dynamic social networks." in IJCAI, 2018, pp. 1632–1638.

[148] L. Wu, L. Chen, R. Hong, Y. Fu, X. Xie, and M. Wang, "A hierarchical attention model for social contextual image recommendation," IEEE TKDE, vol. 32, no. 10, pp. 1854–1867, 2019.

[149] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in SIGIR, 2019, pp. 235–244.

[150] C. Chen, M. Zhang, C. Wang, W. Ma, M. Li, Y. Liu, and S. Ma, "An efficient adaptive transfer neural network for social-aware recommendation," in SIGIR, 2019, pp. 225–234.

[151] Y. Liu, C. Liang, X. He, J. Peng, Z. Zheng, and J. Tang, "Modelling high-order social relations for item recommendation," IEEE TKDE, 2020, preprint.

[152] L. Hu, S. Jian, L. Cao, Z. Gu, Q. Chen, and A. Amirbekyan, "Hers: Modeling influential contexts with heterogeneous relations for sparse and cold-start recommendation," in AAAI , 2019, pp. 3830–3837.

[153] W. Xiao, H. Zhao, H. Pan, Y. Song, V. W. Zheng, and Q. Yang, "Beyond personalization: Social content recommendation for creator equality and consumer satisfaction," in SIGKDD, 2019, pp. 235–245.

[154] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in WWW, 2019, pp. 417–426.

[155] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, and J. Tang, "Session-based social recommendation via dynamic graph attention networks," in WSDM, 2019, pp. 555–563.

[156] Q. Wu, H. Zhang, X. Gao, P. He, P. Weng, H. Gao, and G. Chen, "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in WWW, 2019, pp. 2091–2102.

[157] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, "Diffnet++: A neural influence and interest diffusion network for social recommendation," IEEE TKDE, 2020, preprint.

[158] V. Satuluri, Y. Wu, X. Zheng, Y. Qian, B. Wichers, Q. Dai, G. M. Tang, J. Jiang, and J. Lin, "Simclusters: Community-based representations for heterogeneous recommendations at twitter," in SIGKDD, 2020, pp. 3183–3193.

[159] B. Jin, K. Cheng, L. Zhang, Y. Fu, M. Yin, and L. Jiang, "Partial relationship aware influence diffusion via a multi-channel encoding scheme for social recommendation," in CIKM, 2020, pp. 585–594.

[160] Y. Xian, Z. Fu, S. Muthukrishnan, G. de Melo, and Y. Zhang, "Reinforcement knowledge graph reasoning for explainable recommendation," in SIGIR, 2019, pp. 285–294.

[161] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in AAAI , 2019, pp. 5329–5336.

[162] Y. Cao, X. Wang, X. He, Z. Hu, and T. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in WWW, 2019, pp. 151–161.

[163] X. Wang, X. He, Y. Cao, M. Liu, and T. Chua, "KGAT: knowledge graph attention network for recommendation," in SIGKDD, 2019, pp. 950–958.

[164] K. Zhao, X. Wang, Y. Zhang, L. Zhao, Z. Liu, C. Xing, and X. Xie, "Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs," in SIGIR, 2020, pp. 239–248.

[165] X. Wang, X. He, and T.-S. Chua, "Learning and reasoning on graph for recommendation," in WSDM, 2020, pp. 890–893.

[166] Q. Zhu, X. Zhou, J. Wu, J. Tan, and L. Guo, "A knowledge-aware attentional reasoning network for recommendation," in AAAI , 2020, pp. 6999–7006.

[167] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: a heterogeneous information network approach," in WSDM, 2014, pp. 283–292.

[168] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L. Huang, and C. Xu, "Recurrent knowledge graph embedding for effective recommendation," in RecSys2018, pp. 297–305.

[169] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in SIGKDD, 2017, pp. 635–644.

[170] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for top- N recommendation with A neural co-attention model," in SIGKDD, 2018, pp. 1531–1540.

[171] J. Huang, W. X. Zhao, H. Dou, J.-R. Wen, and E. Y. Chang, "Improving sequential recommendation with knowledge-enhanced memory networks," in SIGIR, 2018, pp. 505–514.

[172] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," in WWW, 2019, pp. 2000–2010.

[173] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang, "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," in SIGKDD, 2019, pp. 968–977.

[174] C.-Y. Tai, M.-R. Wu, Y.-W. Chu, S.-Y. Chu, and L.-W. Ku, "Mvin: Learning multiview items for recommendation," in SIGIR, 2020, pp. 99–108.

[175] C. Chen, M. Zhang, W. Ma, Y. Liu, and S. Ma, "Jointly non-sampling learning for knowledge graph enhanced recommendation," in SIGIR, 2020, pp. 189–198.

[176] R. Sun, X. Cao, Y. Zhao, J. Wan, K. Zhou, F. Zhang, Z. Wang, and K. Zheng, "Multi-modal knowledge graphs for recommender systems," in CIKM , 2020, pp. 1405–1414.

[177] K. Zhou, W. X. Zhao, S. Bian, Y. Zhou, J.-R. Wen, and J. Yu, "Improving conversational recommender systems via knowledge graph based semantic fusion," in SIGKDD, 2020, pp. 1006–1014.

[178] B. Jin, C. Gao, X. He, D. Jin, and Y. Li, "Multi-behavior recommendation with graph convolutional networks," in SIGIR, 2020, pp. 659–668.

[179] L. Hu, S. Xu, C. Li, C. Yang, C. Shi, N. Duan, X. Xie, and M. Zhou, "Graph neural news recommendation with unsupervised preference disentanglement," in ACL, 2020, pp. 4255–4264.

[180] L. Gao, H. Yang, J. Wu, C. Zhou, W. Lu, and Y. Hu, "Recommendation with multi-source heterogeneous information," in IJCAI, 2018, pp. 3378–3384.

[181] J. Gong, S. Wang, J. Wang, W. Feng, H. Peng, J. Tang, and P. S. Yu, "Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view," in SIGIR, 2020, pp. 79–88.

[182] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang, and M. Zhang, "Disenhan: Disentangled heterogeneous graph attention network for recommendation," in CIKM , 2020, pp. 1605–1614.

[183] Z. Han, F. Xu, J. Shi, Y. Shang, H. Ma, P. Hui, and Y. Li, "Genetic meta-structure search for recommendation on heterogeneous information network," in CIKM , 2020, pp. 455–464.

[184] X. Wang, Y. Xu, X. He, Y. Cao, M. Wang, and T.-S. Chua, "Reinforced negative sampling over knowledge graph for recommendation," in WWW, 2020, pp. 99–109.

[185] P. Wang, Y. Fan, L. Xia, W. X. Zhao, S. Niu, and J. Huang, "Kerl: A knowledge-guided reinforcement learning model for sequential recommendation," in SIGIR, 2020, pp. 209–218.

[186] P. Symeonidis, A. Janes, D. Chaltsev, P. Giuliani, D. Morandini, A. Unterhuber, L. Coba, and M. Zanker, "Recommending the video to watch next: an offline and online evaluation at youtv.de," in RecSys2020, pp. 299–308.

[187] W. Lei, G. Zhang, X. He, Y. Miao, X. Wang, L. Chen, and T.-S. Chua, "Interactive path reasoning on graph for conversational recommendation," in SIGKDD, 2020, pp. 2073–2083.

[188] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in CIKM , 2018, pp. 417–426.

[189] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in AAAI , 2015, pp. 2181–2187.

[190] H. Wang, F. Zhang, X. Xie, and M. Guo, "Dkn: Deep knowledge-aware network for news recommendation," in WWW, 2018, pp. 1835–1844.

[191] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, "Learning heterogeneous knowledge base embeddings for explainable recommendation," Algorithms, vol. 11, no. 9, p. 137, 2018.

[192] Z. Wang, G. Lin, H. Tan, Q. Chen, and X. Liu, "Ckan: Collaborative knowledge-aware attentive network for recommender systems," in SIGIR, 2020, pp. 219–228.

[193] W. Meng, D. Yang, and Y. Xiao, "Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation," in SIGIR, 2020, pp. 1091–1100.

[194] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in ICLR, 2016.

[195] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in RecSys2016, pp. 241–248.

[196] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in AAAI , 2019, pp. 346–353.

[197] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in WSDM, 2017, pp. 495–503.

[198] H. Jing and A. J. Smola, "Neural survival recommender," in WSDM, 2017, pp. 515–524.

[199] H. Dai, Y. Wang, R. Trivedi, and L. Song, "Recurrent coevolutionary latent feature processes for continuous-time recommendation," in DLRS, 2016, pp. 29–34.

[200] P. Wu, Y. Tu, Z. Yang, A. Jatowt, and M. Odagaki, "Deep modeling of the evolution of user preferences and item attributes in dynamic social networks," in WWW, 2018, pp. 115–116.

[201] A. Beutel, P. Covington, S. Jain, C. Xu, J. Li, V. Gatto, and E. H. Chi, "Latent cross: Making use of context in recurrent recommender systems," in WSDM, 2018, pp. 46–54.

[202] T. Donkers, B. Loepp, and J. Ziegler, "Sequential user-based recurrent neural network recommendations," in RecSys2017, pp. 152–160.

[203] Q. Wang, H. Yin, Z. Hu, D. Lian, H. Wang, and Z. Huang, "Neural memory streaming recommender networks with adversarial training," in SIGKDD, 2018, pp. 2467–2475.

[204] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, "Sequential recommendation with user memory networks," in WSDM, 2018, pp. 108–116.

[205] E. Smirnova and F. Vasile, "Contextual sequence modeling for recommendation with recurrent neural networks," in DLRS, 2017, pp. 2–9.

[206] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in CIKM , 2017, pp. 1419–1428.

[207] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in DLRS, 2016, pp. 17–22.

[208] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in RecSys2017, pp. 306–310.

[209] P. Loyola, C. Liu, and Y. Hirate, "Modeling user session and intent with an attention-based encoder-decoder architecture," in RecSys2017, pp. 147–151.

[210] F. Yuan, X. He, A. Karatzoglou, and L. Zhang, "Parameter-efficient transfer from sequential behaviors for user modeling and recommendation," in SIGIR, 2020, pp. 1469–1478.

[211] J. Yin, C. Liu, W. Wang, J. Sun, and S. C. Hoi, "Learning transferrable parameters for long-tailed sequential user behavior modeling," in *SIGKDD*, 2020, pp. 359–367.

[212] T. X. Tuan and T. M. Phuong, "3d convolutional networks for session-based recommendation with content features," in *RecSys*, 2017, pp. 138–146.

[213] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *ICDM*, 2018, pp. 197–206.

[214] R. Ren, Z. Liu, Y. Li, W. X. Zhao, H. Wang, B. Ding, and J.-R. Wen, "Sequential recommendation with self-attentive multi-adversarial network," in *SIGIR*, 2020, pp. 89–98.

[215] D. Gligorijevic, J. Gligorijevic, A. Raghuveer, M. Grbovic, and Z. Obradovic, "Modeling mobile user actions for purchase recommendation using deep memory networks," in *SIGIR*, 2018, pp. 1021–1024.

[216] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation," in *IJCAI*, 2019, pp. 3940–3946.

[217] R. Qiu, H. Yin, Z. Huang, and T. Chen, "Gag: Global attributed graph neural network for streaming session-based recommendation," in *SIGIR*, 2020, pp. 669–678.

[218] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, and M. Qiu, "Global context enhanced graph neural networks for session-based recommendation," in *SIGIR*, 2020, pp. 169–178.

[219] Z. Pan, F. Cai, W. Chen, H. Chen, and M. de Rijke, "Star graph neural networks for session-based recommendation," in *CIKM*, 2020, pp. 1195–1204.

[220] T. Chen and R. C.-W. Wong, "Handling information loss of graph neural networks for session-based recommendation," in *SIGKDD*, 2020, pp. 1172–1180.

[221] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention networks," in *IJCAI*, 2018, pp. 3926–3932.

[222] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *SIGIR*, 2015, pp. 403–412.

[223] C. Ma, P. Kang, and X. Liu, "Hierarchical gating networks for sequential recommendation," in *SIGKDD*, 2019, pp. 825–833.

[224] L. Yu, C. Zhang, S. Liang, and X. Zhang, "Multi-order attentive ranking model for sequential recommendation," in *AAAI*, 2019, pp. 5709–5716.

[225] J. Lin, W. Pan, and Z. Ming, "Fissa: fusing item similarity models with self-attention networks for sequential recommendation," in *RecSys*, 2020, pp. 130–139.

[226] L. Wu, S. Li, C.-J. Hsieh, and J. Sharpnack, "Sse-pt: Sequential recommendation via personalized transformer," in *RecSys*, 2020, pp. 328–337.

[227] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, and E. Chen, "Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors," in *SIGKDD*, 2018, pp. 1734–1743.

[228] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *RecSys*, 2017, pp. 130–137.

[229] T. Bai, L. Zou, W. X. Zhao, P. Du, W. Liu, J.-Y. Nie, and J.-R. Wen, "Ctrec: A long-short demands evolution model for continuous-time recommendation," in *SIGIR*, 2019, pp. 675–684.

[230] J. Tang, F. Belletti, S. Jain, M. Chen, A. Beutel, C. Xu, and E. H. Chi, "Towards neural mixture recommender for long range dependent user sequences," in *WWW*, 2019, pp. 1782–1793.

[231] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *CIKM*, 2020, pp. 1893–1902.

[232] J. Wu, R. Cai, and H. Wang, "Déjà vu: A contextualized temporal attention mechanism for sequential recommendation," in *WWW*, 2020, pp. 2199–2209.

[233] W. Ji, K. Wang, X. Wang, T. Chen, and A. Cristea, "Sequential recommender via time-aware attentive memory network," in *CIKM*, 2020, pp. 565–574.

[234] W. Ye, S. Wang, X. Chen, X. Wang, Z. Qin, and D. Yin, "Time matters: Sequential recommendation with complex temporal information," in *SIGIR*, 2020, pp. 1459–1468.

[235] C. Wang, M. Zhang, W. Ma, Y. Liu, and S. Ma, "Make it a chorus: knowledge-and time-aware item modeling for sequential recommendation," in *SIGIR*, 2020, pp. 109–118.

[236] N. Zhu, J. Cao, Y. Liu, Y. Yang, H. Ying, and H. Xiong, "Sequential modeling of hierarchical user intention and preference for next-item recommendation," in *WSDM*, 2020, pp. 807–815.

[237] M. Wang, P. Ren, L. Mei, Z. Chen, J. Ma, and M. de Rijke, "A collaborative session-based recommendation approach with parallel memory modules," in *SIGIR*, 2019, pp. 345–354.

[238] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *WSDM*, 2019, pp. 582–590.

[239] J. Wang, K. Ding, L. Hong, H. Liu, and J. Caverlee, "Next-item recommendation with sequential hypergraphs," in *SIGIR*, 2020, pp. 1101–1110.

[240] C. Ma, L. Ma, Y. Zhang, J. Sun, X. Liu, and M. Coates, "Memory augmented graph neural networks for sequential recommendation," in *AAAI*, 2020, pp. 5045–5052.

[241] X. Guo, C. Shi, and C. Liu, "Intention modeling from ordered and unordered facets for sequential recommendation," in *WWW*, 2020, pp. 1127–1137.

[242] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, and G. Hu, "Ekt: Exercise-aware knowledge tracing for student performance prediction," *IEEE TKDE*, vol. 33, no. 1, pp. 100–115, 2021.

[243] Q. Cui, S. Wu, Q. Liu, W. Zhong, and L. Wang, "Mv-rnn: a multi-view recurrent neural network for sequential recommendation," *IEEE TKDE*, pp. 317–331, 2020.

[244] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *ArXiv*, vol. abs/1410.5401, 2014.

[245] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," in *NeurIPS*, 2015, pp. 2440–2448.

[246] F. Yuan, X. He, H. Jiang, G. Guo, J. Xiong, Z. Xu, and Y. Xiong, "Future data helps training: Modeling future contexts for session-based recommendation," in *WWW*, 2020, pp. 303–313.

[247] Y. Liu, Z. Ren, W.-N. Zhang, W. Che, T. Liu, and D. Yin, "Keywords generation improves e-commerce session-based recommendation," in *WWW*, 2020, pp. 1604–1614.

[248] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *RecSys*, 2017, pp. 130–137.

[249] Q. Wu, Y. Gao, X. Gao, P. Weng, and G. Chen, "Dual sequential prediction models linking sequential recommendation and information dissemination," in *SIGKDD*, 2019, pp. 447–457.

[250] M. F. Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? a worrying analysis of recent neural recommendation approaches," in *Recsys*, 2019, pp. 101–109.

[251] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE TKDE*, vol. 29, no. 1, pp. 17–37, 2016.

[252] J. Gao, M. Galley, and L. Li, "Neural approaches to conversational ai," *ArXiv*, vol. abs/1809.08267, 2019.

[253] Y. Sun and Y. Zhang, "Conversational recommender system," in *SIGIR*, 2018, pp. 235–244.

[254] C. Gao, W. Lei, X. He, M. Rijke, and T.-S. Chua, "Advances and challenges in conversational recommender systems: A survey," *ArXiv*, vol. abs/2101.09459, 2021.

[255] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ArXiv*, vol. abs/1908.09635, 2019.

[256] J. Chen, H. Dong, X. lei Wang, F. Feng, M.-C. Wang, and X. He, "Bias and debias in recommender system: A survey and future directions," *ArXiv*, vol. abs/2010.03240, 2020.

[257] L. Wu, L. Chen, P. Shao, R. Hong, X. Wang, and M. Wang, "Learning fair representations for recommendation: A graph based perspective," in *WWW*, 2021, preprint.

Le Wu is currently an associate professor and Ph.D. supervisor at the Hefei University of Technology (HFUT), China. She received the Ph.D. degree from the University of Science and Technology of China (USTC). Her general area of research interests is data mining, recommender systems and social network analysis. She has published more than 50 papers in referred journals and conferences. Dr. Le Wu is the recipient of the Best of SDM 2015 Award, the Distinguished Dissertation Award from China Association for Artificial Intelligence (CAAI) 2017, and the Youth Talent Promotion Project from China Association for Science and Technology.