SCIENCE CHINA
Information Sciences

• REVIEW •

March 2026, Vol. 69, Iss. 3, 131101:1–131101:32
https://doi.org/10.1007/s11432-024-4568-7

A survey on multi-behavior sequential recommendation

Xiaoqing CHEN[1], Zhitao LI[1], Weike PAN[1*], Qiang YANG[2] & Zhong MING[3,4*]

[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[2]Academy for Artificial Intelligence, Hong Kong Polytechnic University, Hong Kong 999077, China
[3]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518123, China
[4]College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China

Received 23 May 2024/Revised 8 February 2025/Accepted 28 August 2025/Published online 27 January 2026

Abstract People usually have the explicit or implicit desire to get the information they need and are most interested in from massive information, which has led to the creation of personalized recommender systems. Recommender systems are set up to address the issue of information overload in traditional information retrieval systems such as search engines, and have been a significant area of research focusing on recommending information that is of most interest to users. There is a sequential nature to the behaviors of a person interacting with a system, such as examining one item of clothing before examining others. The problem of taking this sequential nature into account to uncover users' interest dynamics in delivering recommendations is known as sequential recommendation (SR). The traditional SR problem merely focuses on a single behavior type of the users, while in real-world scenarios, users tend to engage in multiple types of behaviors, such as examining and adding clothes to the cart before purchasing them. The introduction of multiple behavior types can uncover users' behavior patterns more comprehensively, leading to the proposal of multi-behavior sequential recommendation (MBSR). MBSR considers both sequentiality and heterogeneity of user behaviors, which can achieve state-of-the-art recommendation performance through suitable modeling. Currently, there are some related studies for MBSR, and to the best of our knowledge, there is no related review to introduce and categorize these MBSR studies. Hence, this survey aims to shed light on MBSR, which is a relatively new and worthy direction for in-depth research. First, we introduce MBSR in detail, including its problem definition, application scenarios, and challenges faced. Second, we detail the classification of MBSR methods, including traditional methods and deep learning-based methods, where the former contain neighborhood-based methods and matrix factorization-based methods, and the latter can be classified into different learning architectures based on recurrent neural network (RNN), graph neural network (GNN), Transformer, and generic architectures, as well as architectures that integrate hybrid techniques. In each method, we present related studies from the data perspective and the modeling perspective, analyzing the strengths, weaknesses, and features of these studies, and further conduct experiments on two real-world datasets with classical and recent studies on different methods to show the difference in recommendation performance of these methods. Finally, we discuss some promising future research directions to address the challenges and improve the current status of MBSR.

Keywords multi-behavior sequential recommendation, matrix factorization, RNN, GNN, Transformer

Citation Chen X Q, Li Z T, Pan W K, et al. A survey on multi-behavior sequential recommendation. Sci China Inf Sci, 2026, 69(3): 131101, https://doi.org/10.1007/s11432-024-4568-7

1 Introduction

Nowadays, people are increasingly relying on the Internet to obtain information, and are faced with an information overload due to the complexity and the huge amount of network information. Traditional search engines cannot filter items for each user well, making it difficult for people to quickly access the information they want. As such, recommender systems that can effectively solve the information overload problem and provide personalized services to different users are of great importance. Recommender systems [1, 2] are a fundamental tool to recommend items of most interest to the users from a large amount of information. The recommendation process usually involves collecting and analyzing the users' historical behavior data to learn their preferences and behavior patterns, and thus find the items that better align with their preferences. The historical behavior data used can be divided into explicit feedback and implicit feedback. The explicit feedback data, also known as multi-class feedback, includes behaviors such as a user's ratings and likes on items, while the implicit feedback data, also known as one-class feedback, includes behaviors like examinations, add-to-carts, and purchases.

Since the implicit feedback data are more readily available compared with the explicit feedback data in real-world scenarios, many studies focus on studying recommendation problems based on a single type of implicit feedback behaviors, which brings up the issue of single-behavior recommendation (SBR) [3–5]. However, SBR usually contains

* Corresponding author (email: panweike@szu.edu.cn, mingz@szu.edu.cn)

fewer data, which are prone to the data sparsity or cold-start issues [6]. Besides, there is often more than one type of interaction between users and items in real-world scenarios, such as examination, adding to cart, and purchase in the setting of an e-commerce platform. Since each behavior can represent users' unique preferences or perspectives on items, utilizing heterogeneous behavior data can capture a more comprehensive picture of users' behavior patterns and intentions. As such, researchers turn to the study of multi-behavior recommendation (MBR) problems [6–10]. Different from the SBR problem, MBR provides personalized recommendations to users based on their heterogeneous one-class implicit feedback, including users' target feedback like purchase behaviors and auxiliary feedback such as examinations and favorites.

Since the sequential nature of user interactions can reveal the user's interest dynamics, recent studies separately extend SBR and MBR to single-behavior sequential recommendation (SBSR) [11–15] and multi-behavior sequential recommendation (MBSR) [16–18] by introducing the interaction sequentiality. The MBSR methods introduce both behavior types and sequential information, which can uncover a more comprehensive picture of users' dynamic preferences and intentions through proper modeling, thus bringing more desirable recommendation performance gains than SBSR and MBR methods. Nonetheless, it correspondingly brings more new challenges, including (i) sequence modeling of heterogeneous behavioral feedback, (ii) relationship modeling between user behaviors, (iii) joint long-term and short-term preferences modeling, and (iv) existing noise, bias, and other related issues in MBSR. We discuss the specific challenges in detail in a subsequent section.

To explore how existing studies address the new challenges of MBSR for personalized recommendations, we review some relatively well-known studies and state-of-the-art studies from leading conferences and journals in this survey, hoping to provide some guidance for future research on MBSR. Specifically, we categorize the existing studies on multi-behavior sequential recommendation into traditional methods and deep learning-based methods. Traditional methods consist of neighborhood-based methods and matrix factorization-based methods. For the former, we discuss how to use the neighborhood information to solve the recommendation problem and extend existing methods from SBSR to MBSR, while for the latter, we introduce the general idea of matrix factorization used in recommendation problems and mainly introduce the typical study from the perspective of transfer learning. In deep learning-based methods, we mainly focus on how to apply the ideas of deep learning to the MBSR problem, and describe the MBSR methods based on recurrent neural network (RNN), graph neural network (GNN), Transformer, generic methods, and hybrid methods. Distinct from existing surveys related to sequential recommendation (SR) or MBR [19–21], we further delineate the related studies of each deep learning framework from different data and modeling perspectives. Moreover, we conduct experiments on real-world datasets to compare the recommendation performance of the classical and state-of-the-art methods. Finally, we briefly discuss some future research directions and conclude the paper.

As an emerging field of recommender systems, MBSR lacks specific relevant surveys. In order to provide a comprehensive overview and enable researchers to keep abreast of the latest developments in MBSR, we conduct a survey on this topic, in which we classify and compare various techniques and related studies. To the best of our knowledge, ours is the first study to provide a comprehensive introduction and discussion of MBSR. The key contributions of our survey are summarized as follows.

- We present an in-depth overview of the MBSR problem by discussing its background, problem definition, application scenarios, and existing challenges. Additionally, we provide a comprehensive classification of the current study on MBSR from three key perspectives: technique, data, and modeling.

- We provide a summary of the strengths and weaknesses of each technique employed, alongside a detailed comparison and analysis of representative MBSR studies based on the provided classification.

- We provide experimental validation to compare the recommendation performance of different methods on two real-world datasets.

- We propose valuable future research directions to address the challenges posed by MBSR.

The rest of the paper is organized as follows. In Section 2, we provide the background on MBSR, encompassing four aspects, i.e., problem definition, application scenarios, challenges, and categorization. Then, in Sections 3 and 4, we present an outline of the prominent studies on MBSR in terms of traditional methods and deep learning-based methods, respectively. In Section 5, we conduct experiments on two real-world datasets to explicitly demonstrate the recommendation performance of different MBSR methods. In Section 6, we discuss some possible future research directions for MBSR, and finally, we conclude the paper in Section 7.

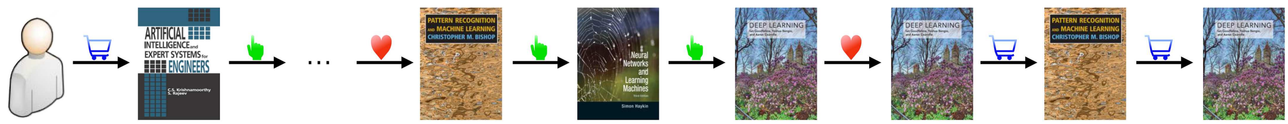

Figure 1 (Color online) Illustration of MBSR.

2 Preliminaries

2.1 Problem definition

The MBSR problem mainly focuses on the next item recommendation in a heterogeneous feedback sequence. We assume that there is a set of users, i.e., $\mathcal{U}$, a set of items, i.e., $\mathcal{I}$, and a set of behaviors (or feedback), i.e., $\mathcal{F}$, in the system. For the corresponding recommendation methods, the input is a set of (user, heterogeneous behavior sequence) pairs, i.e., $\mathcal{D} = \langle u, \mathcal{S}_u \rangle$, where $u \in \mathcal{U}$ represents a user ID, and $\mathcal{S}_u = \{(i_u^1, f_u^1), \ldots, (i_u^t, f_u^t), \ldots, (i_u^{|\mathcal{S}_u|}, f_u^{|\mathcal{S}_u|})\}$ represents the historical interaction sequence between the user u and the items. In the sequence $\mathcal{S}_u$, (i_u^t, f_u^t) represents the (item, behavior) pair composed of the item i and the corresponding behavior f interacted by the user u at the tth time step, where $i_u^t \in \mathcal{I}$ and $f_u^t \in \mathcal{F}$. We use time steps instead of timestamps since MBSR does not generally introduce timestamps which represent precise time in modeling. Properly modeling the input data can help learn the user's preferences, as well as the representations and relationships of the items and behaviors. Based on a typical recommendation method, we can predict the preference value $\hat{r}_{t+1,j}^u$ of the user u for any item $j \in \mathcal{I}$ at the $(t+1)$th time step according to the most recent L historical interactions of user u before the $(t+1)$th time step, i.e., $\mathcal{S}_u^t = \{(i_u^{t-L+1}, f_u^{t-L+1}), \ldots, (i_u^\ell, f_u^\ell), \ldots, (i_u^t, f_u^t)\}$, $t \leqslant |\mathcal{S}_u|$. We can then rank the preference values $\hat{r}_{|\mathcal{S}^u|+1,j}^u$ of the user u for the candidate items $j \in \mathcal{I}$ as follows:

$$\hat{r}_{|\mathcal{S}^u|+1,j}^u = \boldsymbol{g}(j \mid \boldsymbol{f}_u(\mathcal{D})), \tag{1}$$

where $\boldsymbol{f}_u(\mathcal{D})$ denotes the multi-behavior sequential representation of user u obtained after modeling under a given interaction data $\mathcal{D}$, and $\boldsymbol{g}(\cdot)$ denotes a function that measures the relationship between user u and candidate item j, such as the dot product. Then, we can generate a top-K list of items for user u, which indicates the next items that user u is most likely to interact with. We illustrate the general MBSR process in Figure 1, and show the commonly used symbols and corresponding interpretations in Table 1, where we employ various font styles to denote diverse types of notations, i.e., uppercase bold for matrices, lowercase bold for vectors, lowercase non-bold for scalars, and copperplate for sets.

2.2 Application scenarios

In recommender systems, MBSR is a relatively new research hotspot, attracting extensive attention from both academia and industry. In the industry, the related studies on MBSR are mainly applied to the click-through rate (CTR) prediction tasks. CTR leverages historical (user, item) interaction information to generate a list of items for recommendation to the user at the next time step. However, in contrast to SR in academia, the CTR task ranks items by predicting user clicks on them, and tends to introduce more side information from users, items, and platforms into the modeling, as well as strictly limits on the online latency rate. Thus, the data processing methods and models used tend to be different from those in SR. Currently, MBSR is commonly encountered in many areas, varying from e-commerce [22, 23] and video recommendation [24, 25] to news recommendation [26, 27]. In e-commerce, researchers predict the items that users are most likely to purchase by analyzing the multi-behavior sequential information of their examinations, add-to-carts, favorites, and purchases [16, 28, 29]. In video recommendation, users generate behaviors like examinations, shares, and others, where the sharing behavior can be used as the target behavior type, and the examination behavior can be used as the auxiliary behavior type [30]. In news recommendation, users may interact with news by explicit feedback (e.g., dislike) and implicit feedback (e.g., examination), allowing for the incorporation of this feedback to infer their positive and negative preferences [31].

To better learn the real preferences of users, some research studies also take into account the dwelling time, category information, and other fine-grained information in the MBSR problem [32–34]. Moreover, some studies on MBSR focus not only on better capturing user preferences, but also on designing relevant multi-behavior sequential recommendation algorithms in the context of user privacy protection issues. Importantly, MBSR methods also consider the heterogeneous behavior information of users and the sequential information within or among the behaviors, allowing for more useful information to be learned when modeling, which makes them closer to real

Table 1 Some notations and explanations.

Notation	Explanation
n	User number
m	Item number
$\mathcal{R}$	The observed set of (user, item, behavior) tuples
$\mathcal{U}$	User set
$\mathcal{I}$	Item set
$\mathcal{F}$	Behavior set
$u \in \mathcal{U}$	User ID
$i \in \mathcal{I}$	Item ID
$f \in \mathcal{F}$	Behavior ID
$\mathcal{S}_u$	The sequence of (item, behavior) pairs that user u has interacted with
$\mathcal{S}_{\mathrm{e}}$	The sequence examined by user u
$\mathcal{S}_{\mathrm{n}}$	The sequence unexamined by user u
$\mathcal{S}_{\mathrm{l}}$	The sequence that user u has liked
$\mathcal{S}_{\mathrm{d}}$	The sequence that user u has disliked
$i_u^t \in \mathcal{I}$	The item interacted by user u at the time step t ($t \in \{1, 2, \ldots, \lvert\mathcal{S}_u\rvert\}$)
$f_u^t \in \mathcal{F}$	The behavior of user u at the time step t ($t \in \{1, 2, \ldots, \lvert\mathcal{S}_u\rvert\}$)
$U_u \in \mathbb{R}^{d\times 1}$	The embedding of user u
$V_i \in \mathbb{R}^{d\times 1}$	The embedding of item i
$F_f \in \mathbb{R}^{d\times 1}$	The embedding of behavior f
$\hat{r}_{t,i}^u$	The predicted preference of user u to item i at the time step t
$\hat{r}_{t,i,f}^u$	The predicted preference that user u generates behavior f on item i at the time step t
$\mathrm{Em}(\cdot)$	The ID-to-embedding function
$\sigma(\cdot)$	The sigmoid function
$\odot$	The element-wise product function

recommendation scenarios. Hence, it is of great significance to design a recommendation algorithm for multi-behavior sequential recommendation.

2.3 Challenges

The MBSR problem involves modeling both multiple behaviors and behavior sequences, contributing to the necessity to consider the existing problems of MBR and SBSR, as well as how to integrate these two kinds of information well. We present the main challenges of MBSR, and give a brief overview of how current studies address these challenges. More details for each study can be found in Sections 3 and 4.

- Sequence modeling of heterogeneous behavioral feedback. In a traditional sequential recommendation problem [12, 14], researchers mostly consider only a single type of behavior, ignoring the potential and importance of other behaviors, especially in instances where the utilized data for the target behaviors is sparse. It indicates that it is necessary to model the users' multiple heterogeneous behaviors in the sequential recommendation problem. However, different from SBSR, the uncertainty of users' intention due to heterogeneous behaviors makes it more challenging to predict the users' preferences in MBSR. Hence, it is a key and challenging issue to model heterogeneous behaviors well in sequential recommendations without information loss. Existing studies model heterogeneous behavior sequences in the following ways: modeling multiple behavior-specific item subsequences separately and then fusing the output representations like DyMus [35] and multi-relational graph neural network model for session-based target behavior prediction (MGNN-SPred) [30], encoding behavior types at the embedding layer and fusing them with item representations as the input of sequential modeling like recurrent log-bilinear model (RLBL) [16] and micro-behaviors and item knowledge into multi-task learning for session-based recommendation (MKM-SR) [17], and distinguishing and fusing sequential representations of different behavior types by some specific settings like behavior-specific channels in the network such as multi-behavior sequential Transformer recommender (MB-STR) [18] and PBAT [36].
- Relationship modeling between user behaviors. In the MBSR problem, multiple behaviors of users are often related to each other [32]. For example, on an e-commerce platform, users tend to examine an item and check the reviews of the item before purchasing it, or purchase an item after examining and adding to the cart other items of the same category. Different from MBR, which does not consider the sequential relationship of behaviors, MBSR takes the sequential nature of various behaviors into account. For example, MBR treats both cases the same for

users who examine first and then purchase and for users who purchase and then examine, whereas MBSR considers the distinction between the two in modeling. According to the above issues mentioned, there are correlations and transitions among different behaviors within a user-item interaction sequence, which is a great challenge in modeling. Previous methods in modeling user behavior relevance mainly include the following: (i) modeling behavior transition patterns, such as setting up behavior transition matrices [37], constructing behavior-relation distributions [36], and aggregating node-neighborhood representations based on different types of behavior transition [30, 38–40]; (ii) introducing the target behavior to capture the contribution of different historical behaviors [41, 42]; and (iii) capturing the information similarities and differences of different behavior types through contrastive learning, multi-task learning, and partial network parameter sharing [18, 43, 44].

- Joint long-term and short-term preference modeling with heterogeneous behaviors of users. Most of the traditional recommendation algorithms statically model the interaction information between users and items [45–47], which usually reveals users' long-term stable preferences. However, they ignore the dynamic changes of users' sequential behaviors in interactions with items. The dynamics of user preferences indicate the user's current short-term preferences [48], which can be revealed in the dynamically changing behavioral sequential information of the user. Since the user's interactive behavior information is a behavior sequence that naturally evolves over time, the sequential information can dynamically display the user's long-term stable preferences and short-term needs. How to take the sequential information into account is the main challenge of SBSR. However, compared with SBSR, the behavior heterogeneity of MBSR leads to an even greater challenge in modeling a user's long-term and short-term preferences simultaneously. Previous studies essentially split a user sequence into multiple subsequences, model short-term preferences for each subsequence, and then aggregate the subsequence representations as the long-term interests, where the subsequences can be of a particular length [16, 49, 50] or of a particular behavior type [51].
- Related issues such as noise and bias. Some previous studies regard unexamined behaviors and missing behaviors as implicit negative feedback of users, or simply ignore them [52–54]. However, unexamination does not always represent a negative user preference [55], nor does an examination represent a positive user preference, where there may be inadvertent examinations. As for bias, since most current recommender systems tend to utilize the implicit feedback of users to make recommendations to them with the goal of more accurate item ranking, there may be a selection bias in implicit feedback data (e.g., a user may examine on an item simply because it is ranked highly). Currently, research on bias and noise remains limited. DMT [29] addresses implicit feedback bias using a bias deep neural network, while some studies tackle implicit feedback noise through attention networks [31, 56] or feature orthogonal mapping [57]. As such, the possible noise and bias in MBSR deserve more attention.

2.4 Categorization

In accordance with widely adopted techniques in recommender systems [2, 21] and considering the distinct technical characteristics among various approaches, methods employed for MBSR can be categorized at the technical level into two paradigms: traditional methods and deep learning-based methods. Traditional methods primarily encompass neighborhood-based methods and matrix factorization-based methods. With the continuous rise of artificial neural networks and deep learning in various fields in recent years, alongside the increasing complexity of information (i.e., the necessity to model heterogeneous behavior information and behavioral sequential information simultaneously), the majority of the current studies utilize deep learning-based methods for modeling to achieve higher recommendation accuracy, while less emphasis is placed on traditional methods. As a result, we primarily pay attention to the deep learning-based methods for the MBSR problem in this paper.

For deep learning-based methods, we review classical, state-of-the-art, and recent studies categorized by their neural network architectures, including RNN-based, GNN-based, Transformer-based, generic-method-based, and hybrid-method-based learning architectures, where the generic-method-based learning architecture with a flexible framework can encompass any SR method, and the hybrid-method-based learning architecture makes a strategic integration of complementary techniques to leverage their respective advantages. First, we introduce the basic paradigm of each type of neural network architecture, and then discuss the related studies applied to the MBSR problem. In these studies, the loss functions used generally contain pointwise-based loss functions, such as logistic loss [58] and square loss [59], and pairwise-based loss functions like BPR loss [5], as well as their combined or varied loss functions. Second, we classify the related studies according to different data perspectives and modeling perspectives under each neural network architecture, as shown in Figure 2. Specifically, the data perspectives comprise four forms, namely a sequence of (item, behavior) pairs, some behavior-specific subsequences of items, a behavior-agnostic sequence of items, and a sequence of behaviors; the modeling perspectives include both local and global approaches; and some of the MBSR studies may integrate different data or modeling perspectives. As for related studies with each neural network architecture, we discuss the strengths, weaknesses, features, and

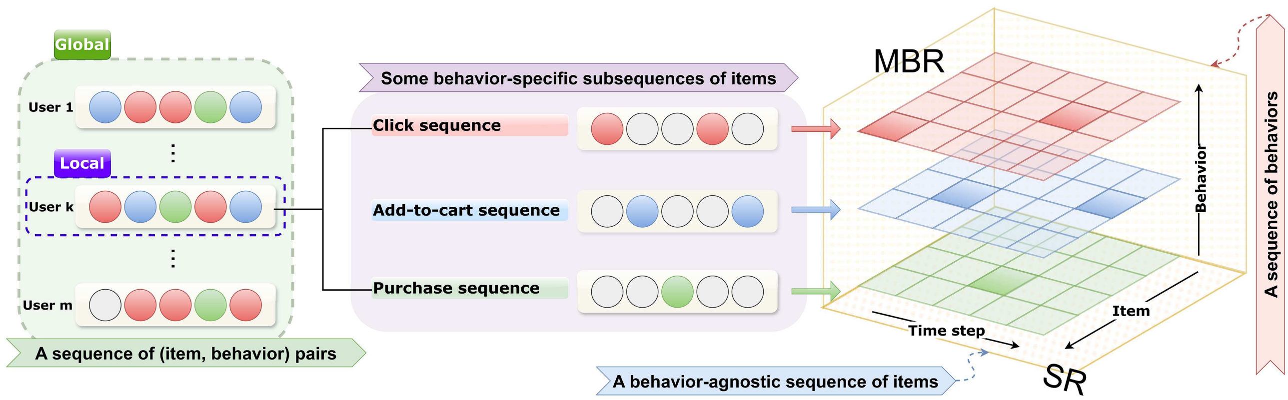

Figure 2 (Color online) MBSR architecture in the fine-grained data and modeling perspectives. In terms of the modeling perspective, methods for MBSR can integrate all user behavior sequences to capture global user preferences and behavior patterns, or model locally for a single user sequence to capture personalized interaction patterns. When modeling each user behavior sequence, there are four forms of data inputs, i.e., a sequence of (item, behavior) pairs, some behavior-specific subsequences of items, a behavior-agnostic sequence of items, and a sequence of behaviors. For each user behavior sequence, the MBSR architecture considers information in the dimensions of item, behavior type, and time step, and reduces to MBR and SR when omitting the behavior information and the sequential information, respectively.

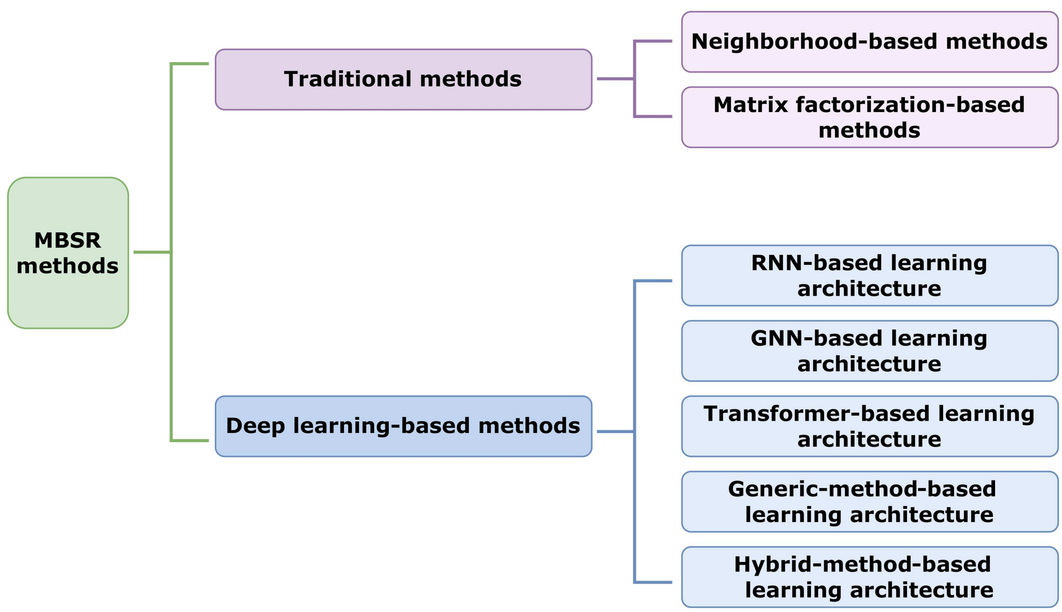

Figure 3 (Color online) Categorization of MBSR methods.

some issues related to sequential heterogeneous information, such as considering more fine-grained information in modeling (e.g., item category), and taking into account how to address noise and bias. We illustrate the above detailed categorization in Figure 3, and further conduct experiments on two real-world datasets to illustrate the effectiveness of some representative studies from different branches.

3 Traditional methods

3.1 Neighborhood-based methods

The neighborhood-based method [60] is an early method in recommender systems, which is mainly divided into user-based collaborative filtering, item-based collaborative filtering, and hybrid collaborative filtering. In the idea of user-based collaborative filtering, two users who have similar tastes in the past may also have similar tastes in the future, while in the idea of item-based collaborative filtering, users may purchase items similar to those purchased in the past. Hybrid collaborative filtering combines the ideas of the first two, and its prediction is a weighted combination of them. Regardless of the form of neighborhood-based methods, the main core concept is similarity.

However, defining the similarity is a matter of concern for MBSR.

3.1.1 *Basic paradigm*

In the case of implicit feedback data, the similarity between two items can be calculated using measures such as the Jaccard index and the cosine similarity. Taking item-based collaborative filtering with the Jaccard index as an example, the similarity between items k and j is calculated as follows:

$$s_{ii'} = \frac{|\mathcal{U}_i \cap \mathcal{U}_{i'}|}{|\mathcal{U}_i \cup \mathcal{U}_{i'}|}, \tag{2}$$

where $\mathcal{U}_i$ and $\mathcal{U}_{i'}$ denote the set of users who have interacted with item i and item i', respectively.

Based on the calculated similarity, we may select the top-K nearest item set $\mathcal{N}_{i'}$ for each item i', and then predict the score according to the following formula:

$$\hat{r}_{ui'} = \sum_{i \in \mathcal{I}_u \cap \mathcal{N}_{i'}} s_{ii'}. \tag{3}$$

As the predicted rating of an item increases, the possibility that the user will be interested in it increases accordingly. Although there is almost no work on MBSR using neighborhood-based methods, we introduce the bidirectional item similarity (BIS) [12] for SBSR, to illustrate the idea of the use of similarity in sequential recommendation. It is expected to have some possibilities and inspirations to solve the MBSR problem.

3.1.2 *BIS*

BIS designs a bidirectional item similarity to perform the next-item recommendation task. The bidirectional item similarity between items i and i' is defined as follows:

$$\mathrm{sim}_{i' \to i}^{(\ell,\rho)} = \frac{\sum_{u \in \mathcal{U}_i \cap \mathcal{U}_{i'}} \delta\left(-\rho\ell \leqslant (t - t') \leqslant \ell\right)}{|\mathcal{U}_i \cup \mathcal{U}_{i'}|}, \tag{4}$$

where ℓ and ρ are hyperparameters. In this equation, if the condition $-\rho\ell \leqslant (t-t') \leqslant \ell$ is satisfied, $\delta(-\rho\ell \leqslant (t-t') \leqslant \ell)$ will be set to 1, so that the numerator will be added by 1 and the similarity between items i and i' will increase accordingly. It is worth noting that when ρ is equal to 1 and $\ell \to \infty$, the bidirectional item similarity degenerates to Jaccard index, i.e., $\mathrm{sim}_{i' \to i}^{(\infty,1)} = \frac{|\mathcal{U}_i \cap \mathcal{U}_{i'}|}{|\mathcal{U}_i \cup \mathcal{U}_{i'}|}$, which does not take into account any sequential information. When predicting the preference score, BIS only considers the bidirectional item similarities of the last k items that user u has interacted with.

Obviously, BIS and ABIS (adaptive BIS) [12], an improved version of BIS based on some factorization techniques, can be extended to solutions for MBSR. For example, if we divide the input sequence into multiple behavior-specific subsequences, we can easily apply BIS and ABIS for each subsequence. These neighborhood-based methods are easy to maintain and more interpretable, but they are less able to capture user preferences and lack transitivity, which means that two users will never be connected if they have not bought a common item. Moreover, ABIS only considers the closest neighboring items in modeling, ignoring users' long-term preferences and periodicity.

3.2 Matrix factorization-based methods

Although neighborhood-based methods may provide interpretability, their aforementioned disadvantages and lower efficiency make them less applicable to MBSR. To address the problem of non-transitivity, matrix factorization is proposed to connect users who have not purchased common items before [61, 62]. There are also some SR studies based on matrix factorization, including FPMC [11] and TransRec [63] for SBSR, and TransRec++ [37] for MBSR. We can similarly extend FPMC and TransRec to MBSR versions simply by dividing a user sequence into multiple behavior-specific subsequences. We will first introduce the basic paradigm of matrix factorization in recommender systems, and then introduce TransRec++, which brings the idea of behavior transition on top of TransRec.

3.2.1 *Basic paradigm*

In recommender systems, the idea of matrix factorization is mainly reflected in transforming the (user, item) interaction matrix into the inner product of two low-rank matrices, i.e., a user-specific matrix and an item-specific matrix. Taking the rating matrix $\boldsymbol{M} \in \mathbb{R}^{m \times n}$ formed by (user, item) interactions as an example, $\boldsymbol{M}$ is decomposed

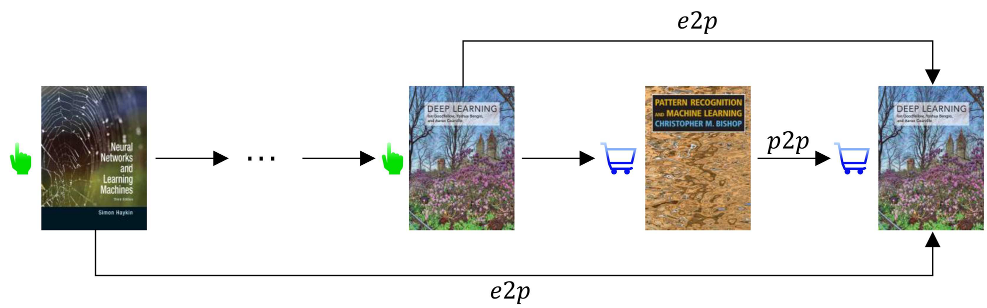

Figure 4 (Color online) Illustration of behavior transition.

into a user matrix $\boldsymbol{U} \in \mathbb{R}^{m\times k}$ and an item matrix $\boldsymbol{V} \in \mathbb{R}^{n\times k}$, so that each missing value (i.e., a predicted value) $\hat{r}_{ui}$ in the rating matrix can be obtained by multiplying the user embedding U_u and the item embedding V_i:

$$\hat{r}_{ui} = U_u \cdot V_i^{\top}. \tag{5}$$

3.2.2 *TransRec++*

TransRec++ [37] introduces several behavior transition vectors to capture the sequential relationships between user behaviors and their dynamics, and takes into account some recent preceding items which can learn the weights automatically. The behavior transition vectors include four types, i.e., from examination to examination $e2e$, from examination to purchase $e2p$, from purchase to examination $p2e$, and from purchase to purchase $p2p$, which we illustrate in Figure 4.

In step ℓ, the overall translation vector of user u to the target item i_u^t is calculated by the following equation:

$$\tilde{U}_u^{(\ell)i_u^t} = U_u + U_u^{(\ell)b(i_u^{t-\ell})2b(i_u^t)}, \tag{6}$$

where $b(\cdot)$ denotes the behavior type. To achieve a transition of item $i_u^{t-\ell}$ to a future item i_u^t in step ℓ for a user's sequence, the formula can be calculated as follows:

$$V_{i_u^{t-\ell}} + \tilde{U}_u^{(\ell)i_u^t} \approx V_{i_u^t}, \quad \ell = 1, 2, \ldots, L, \tag{7}$$

where $V_{i_u^{t-\ell}}$ and $V_{i_u^t}$ are the embedding vectors of item $i_u^{t-\ell}$ and item i_u^t, respectively. The prediction formula is defined as follows:

$$\hat{r}_{ui_u^t} = p_{i_u^t} - \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u) \,||\, V_{i_u^{t-\ell}} + \tilde{U}_u^{(\ell)i_u^t} - V_{i_u^t} \|_2^2, \tag{8}$$

where $p_{i_u^t}$ is an item bias, and η_ℓ, η_ℓ^u denote a global weight and a user-specific weight, respectively.

As one of the few matrix factorization-based solutions towards the next-item recommendation in MBSR, TransRec++ combines the ideas of Fossil [64] and TransRec [63] to address behavior heterogeneity well. The proposed behavior transition can also be utilized in other deep learning-based approaches to reach better performance, such as RIB [32] and behavior-intensive neural network (BINN) [28] that we will mention later. However, TransRec++ becomes more complex in modeling when there are more behavior types, and when it only contains two types of behaviors, its time complexity is already five times that of the SBSR-oriented method TransRec. There may also be noise in the modeling of behavior transition, e.g., $e2e$ may be caused by the user's inadvertent examination.

In summary, as a conventional approach, the matrix factorization-based recommendation algorithm has several benefits, including high interpretability and computational efficiency. These algorithms employ a linear model, which possesses a straightforward structure, and a clear association between the modeling concept and the problem under consideration, leading to a higher degree of interpretability. The algorithms are computationally efficient as the model has few parameters, and typically, only matrix multiplication operations are necessary for computation. However, matrix factorization-based recommendation algorithms face challenges in handling non-linear features like sequential information and higher-order neighborhood information.

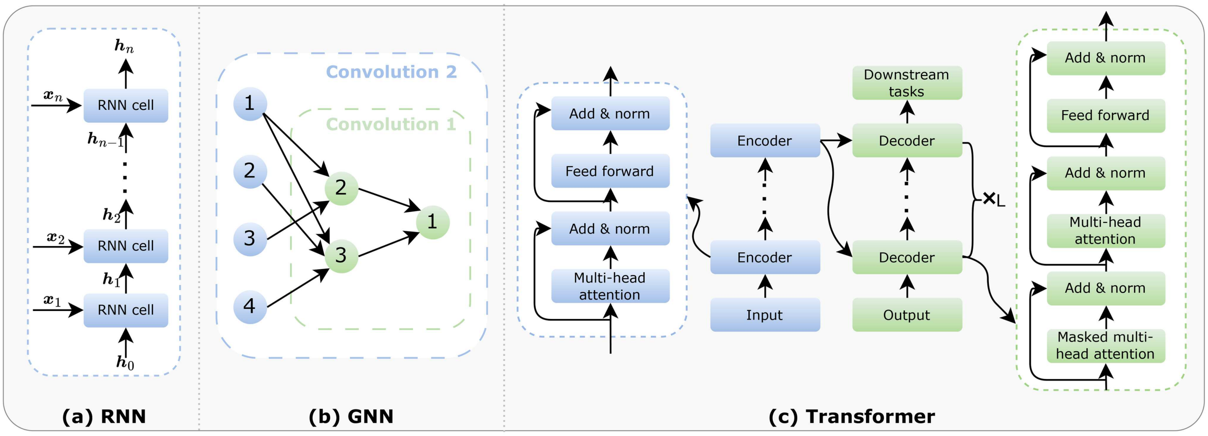

Figure 5 (Color online) Illustration of mainstream deep learning architectures for MBSR.

4 Deep learning-based methods

Due to the insignificant improvement in the recommendation effectiveness of matrix factorization-based methods, researchers have turned to studying deep learning-based algorithms. Deep learning [65, 66] is an improvement on the traditional neural network, and a multi-layer perceptron (MLP) with multiple hidden layers is a typical deep learning architecture. Deep learning has made substantial progress in a variety of application areas, including natural language processing and generation [67, 68], speech recognition and synthesis [69], as well as computer vision [70, 71]. Recently, deep learning has gained increasing use in recommender systems, demonstrating high recommendation performance in multi-behavior recommendation [72, 73], sequential recommendation [13, 14], and federated recommendation [74, 75].

In MBSR, the deep learning networks primarily employed are RNN, GNN, and Transformer, as shown in Figure 5. RNN captures sequential information by maintaining a hidden state and processing inputs sequentially. It performs well in sequence modeling with relatively low computational complexity, but faces the problems of gradient disappearance and gradient explosion, which limit its ability to model long sequences. Additionally, the output at any given time depends on the computations and outputs from the previous time step, resulting in inefficiency and difficulty in predicting future sequence information. Furthermore, RNN struggles to explicitly model the complex relationships between different behavior types. In contrast, GNN leverages directed graphs to capture item transition relationships within sequences, and aggregates nodes and their neighbors' information through a message-passing mechanism to capture interactions among multiple behaviors in sequences. GNN can capture dependencies of different behavior types and effectively cope with data sparsity, but it mainly focuses on the graph structure information, and is weak in modeling temporal dependencies within sequences. Additionally, as the number of behavior types increases, the complexity of the graph structure may rise, leading to increased computational complexity. Transformer utilizes self-attention mechanisms to capture sequence information, allowing it to weigh the importance of all elements in the sequence simultaneously, which can effectively model both local and global dependencies. This enables effective modeling of both local and global dependencies. Although Transformer has higher computational complexity than the aforementioned networks, it is able to capture long-distance dependencies in sequences, and possesses superior parallel computation capabilities, scalability, and interpretability. As a result, Transformers often demonstrate outstanding performance in MBSR tasks involving long sequences. Moreover, designing hybrid models that combine the strengths of different approaches, such as GNNs and Transformers, holds the potential to further enhance performance.

As such, in this section, we mainly discuss the applications and the corresponding studies of deep learning in MBSR in terms of different neural network architectures (i.e., RNN, GNN, Transformer, generic methods, and hybrid methods). We present how different studies model sequentiality and heterogeneity in MBSR, and examine whether these studies address any other specific challenges. Additionally, we distinguish between the different applications of these studies, such as next-item, next-basket, and session-based recommendation, and discuss their features, strengths, and weaknesses. This helps to establish a better understanding of the use of deep learning techniques in MBSR.

4.1 RNN-based learning architecture

4.1.1 *Basic paradigm*

RNN [76] is a classical deep learning method that can effectively process data with sequentiality. Currently, RNN has been applied to numerous fields, including information retrieval, speech recognition, and machine translation. Since RNNs can take into account the characteristics of sequences, they have also been utilized to solve the SBSR and MBSR problems in early studies [14, 16, 32].

RNN contains multiple RNN cells, with its basic structure illustrated in Figure 5(a). In the RNN learning architecture, the current time step receives the output of the previous time step as the input, and the output obtained from the RNN cell will be used as the input of the next time step, so as to capture the sequential nature of the data. Each cell of the RNN is a layer of a deep feedforward neural network, and a set of learning parameters is shared across different time steps to capture sequential features and reduce the model complexity. The basic formula of the RNN is as follows:

$$\boldsymbol{h}_t = \sigma\left(\boldsymbol{W}_{\mathrm{xh}}\boldsymbol{x}_t + \boldsymbol{W}_{\mathrm{hh}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\mathrm{h}}\right), \tag{9}$$

$$\hat{\boldsymbol{y}}_t = \sigma\left(\boldsymbol{W}_{\mathrm{ho}}\boldsymbol{h}_t + \boldsymbol{b}_{\mathrm{o}}\right), \tag{10}$$

where $\boldsymbol{W}_{\mathrm{xh}} \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_{\mathrm{hh}} \in \mathbb{R}^{d\times d}$, and $\boldsymbol{W}_{\mathrm{ho}} \in \mathbb{R}^{d\times d}$ are the corresponding weight matrices, and $\boldsymbol{b}_{\mathrm{h}}$ and $\boldsymbol{b}_{\mathrm{o}}$ are the corresponding bias vectors.

However, there is a certain challenge in the training process of RNNs, i.e., as the depth deepens, RNNs have the problem of gradient disappearance or gradient explosion, and thus they are prone to difficulties in dealing with the long-term dependency of data [77]. To address this issue, many derivative methods based on RNNs have been proposed, among which the most well-known ones are long short-term memory (LSTM) [78] and gated recurrent unit (GRU) [79], a simplified version of LSTM. Both LSTM and GRU set up a hidden unit in the hidden layer to store long-term features, which enables them to address the issue of modeling long-term data dependency.

4.1.2 *Methods in MBSR*

There are some research on RNN-based neural network architectures for solving MBSR problems, which differ in terms of the perspective of the input sequences and the perspective of modeling the behavior types. Specifically, from the data perspective, most of the studies have an input sequence of (item, behavior) pairs, such as RLBL [16], RIB [32], BINN [28], AIR [80], HUP [33], intention-aware recommender system (IARS) [34], and MAINT [81]. In contrast, other studies have some behavior-specific subsequences of items, such as CBS [82], DIPN [83], DeepRec [84], multi-behavior network (MBN) [44], and DyMus [35]. From the modeling perspective, DeepRec models a user's behavior types in the cloud from a global perspective and in the user's own client from a local perspective, while other studies mentioned above utilize a local perspective only to model the behavior types. We distinguish and summarize these studies in Table 2 and describe some of them in detail as shown below.

RLBL. The RLBL [16] is the first work oriented towards next-item recommendation. RLBL integrates the ideas of RNN and log-bilinear (LBL) to address the challenge of long-term and short-term preference modeling. Specifically, RLBL uses behavior-specific transition matrices to distinguish between heterogeneous behaviors in a user's historical interaction sequence and splits the sequence into multiple windows. Then, RLBL captures the short-term contextual information for each window by LBL, and finally integrates these features at the granularity of the window by RNN to construct the user's long-term contextual information.

In RLBL, each window contains a sequence of (item, behavior) pairs of length n, i.e., $\{(i_u^{t-n+1}, f_u^{t-n+1}), \ldots, (i_u^t, f_u^t)\}$. In the pair (i_u^{t-i}, f_u^{t-i}) of the sequence, RLBL uses an item embedding $V_{i_u^{t-i}} \in \mathbb{R}^{d\times 1}$ to represent the historically interacted item i_u^{t-i} of user u, a behavior correlation embedding $\boldsymbol{M}_{f_u^{t-i}} \in \mathbb{R}^{d\times d}$ to represent the user's behavior f_u^{t-i} for item i_u^{t-i}, and a position transition embedding $\boldsymbol{C}_i \in \mathbb{R}^{d\times d}$ to separately capture the position context information of each position in the window $\left(i_u^{t-i}, f_u^{t-i}\right), i \in \{0, 1, \ldots, n-1\}$. Hence, the hidden state $\boldsymbol{h}_{t+1}$ at the $(t+1)$th time step is calculated as follows:

$$\boldsymbol{h}_{t+1} = \boldsymbol{W}_{\mathrm{RLBL}}\boldsymbol{h}_{t-n+1} + \sum_{i=0}^{n-1} \boldsymbol{C}_i \boldsymbol{M}_{f_u^{t-i}} V_{i_u^{t-i}}, \tag{11}$$

where $\boldsymbol{W}_{\mathrm{RLBL}} \in \mathbb{R}^{d\times d}$ is utilized to capture the sequential information between the hidden state $\boldsymbol{h}_{t+1} \in \mathbb{R}^{d\times 1}$ and the hidden state $\boldsymbol{h}_{t-n+1} \in \mathbb{R}^{d\times 1}$. And then the predicted preference that user u generates behavior f on item i at

Table 2 Data & modeling perspectives and features used in studies based on RNN learning architecture.

Studies	Data perspective	Model perspective	Features
RLBL [16]	A sequence of (item, behavior) pairs	Local	Model both short-term and long-term preferences; capture the influence of heterogeneous behaviors by utilizing a behavior correlation matrix
RIB [32]	A sequence of (item, behavior) pairs	Local	Leverage GRU and attention mechanism simultaneously
BINN [28]	A sequence of (item, behavior) pairs	Local	Design the CLSTM and the Bi-CLSTM, where the behavior vector is as the context in the LSTM
CBS [82]	Some behavior-specific subsequences of items	Local	Design of models with and without shared parameters for behaviors simultaneously, towards the next-basket recommendation
DIPN [83]	Some behavior-specific subsequences of items	Local	Leverage GRU and attention mechanism simultaneously; behaviors are specific, including swipe, touch, and browse interactive behavior
AIR [80]	A sequence of (item, behavior) pairs	Local	Design an attentional RNN to model the user's intention transitions, introduce the category attribute
HUP [33]	A sequence of (item, behavior) pairs	Local	Design the Behavior-LSTM, which adds a behavior gate and a time gate to the LSTM, leverages the attention mechanism, and introduces the category attribute
IARS [34]	A sequence of (item, behavior) pairs	Local	Propose Soft-MGRU (a multi-behavior gated recurrent unit) with sharing parameters among behaviors, leverage attention mechanism; introduce the category attribute
DeepRec [84]	Some behavior-specific subsequences of items	Local + Global	Utilizing multi-behavior sequence data to make privacy-preserving recommendations
MBN [44]	Some behavior-specific subsequences of items	Local	The overall Meta-RNN and the separate Behavior-RNN share the learned potential representations by gathering and then scattering towards the next-basket recommendation
MAINT [81]	A sequence of (item, behavior) pairs and a purchase-specific subsequence of items	Local	Capture the user's multifaceted intent through the target behavior sequence, leverage attention mechanism; introduce the category attribute
DyMuS [35]	Some behavior-specific subsequences of items	Local	Capture both sequence-level and item-level dynamic correlations through dynamic routing

the $(t+1)$th time step is calculated as follows:

$$\hat{r}_{t+1,i,f} = (\boldsymbol{h}_{t+1} + U_u)^{\mathrm{T}} \boldsymbol{M}_f V_i, \tag{12}$$

where $U_u \in \mathbb{R}^{d\times 1}$ is the user embedding, and $\boldsymbol{h}_{t+1} \in \mathbb{R}^{d\times 1}$ is the representation incorporating the long-term and short-term preferences of user u.

RLBL and its extended version TA-RLBL [16], which considers continuous time differences, can model the short-term context information well with the consideration of the sequential and heterogeneous nature of user behaviors by RNN and behavior-specific matrices, respectively. However, the modeling of user behaviors is relatively straightforward, and there are some important issues that are overlooked. For example, the transition matrix is the same for all users, and it does not take into account the feature information of the items.

RIB. An interpretable recommendation framework from the micro behavior perspective (RIB) [32], another classic work towards next-item recommendation, models heterogeneous behaviors and dwell time to capture more fine-grained user information using GRU. Specifically, RIB takes a sequence of (item, behavior) pairs as input, taking items and behaviors encoded as item embeddings and behavior embeddings via an embedding layer, respectively. Then the embedding $\boldsymbol{e}_t \in \mathbb{R}^{2d\times 1}$ is obtained by concatenating the above two embeddings and fed into a GRU layer to obtain the hidden state at each time step. The calculation equations of the reset gate $\boldsymbol{r}_t \in \mathbb{R}^{d\times 1}$, the update gate $\boldsymbol{z}_t \in \mathbb{R}^{d\times 1}$, the internal state $\boldsymbol{c}_t \in \mathbb{R}^{d\times 1}$ and the external state $\boldsymbol{h}_t \in \mathbb{R}^{d\times 1}$ at the tth time step in GRU are shown below:

$$\boldsymbol{r}_t = \sigma\left(\boldsymbol{W}_{\mathrm{er}}\boldsymbol{e}_t + \boldsymbol{W}_{\mathrm{hr}}\boldsymbol{h}_{t-1}\right), \tag{13}$$

$$\boldsymbol{z}_t = \sigma\left(\boldsymbol{W}_{\mathrm{ez}}\boldsymbol{e}_t + \boldsymbol{W}_{\mathrm{hz}}\boldsymbol{h}_{t-1}\right), \tag{14}$$

$$\boldsymbol{c}_t = \tanh\left(\boldsymbol{W}_{\mathrm{ec}}\boldsymbol{e}_t + \boldsymbol{W}_{\mathrm{hc}}\left(\boldsymbol{r}_t \cdot \boldsymbol{h}_{t-1}\right)\right), \tag{15}$$

$$\boldsymbol{h}_t = \left(1 - \boldsymbol{z}_t\right)\boldsymbol{h}_{t-1} + \boldsymbol{z}_t\boldsymbol{c}_t, \tag{16}$$

where $\boldsymbol{W}_{\mathrm{er}}$, $\boldsymbol{W}_{\mathrm{ez}}$, $\boldsymbol{W}_{\mathrm{ec}} \in \mathbb{R}^{d\times 2d}$, and $\boldsymbol{W}_{\mathrm{hr}}$, $\boldsymbol{W}_{\mathrm{hz}}$, $\boldsymbol{W}_{\mathrm{hc}} \in \mathbb{R}^{d\times d}$ are the learnable weight parameters inside GRU. Then the hidden state is passed into an attention layer to get the attention score for each time step. Finally, in the output layer, the hidden states of each time step are multiplied with the corresponding attention scores, where the results are added to obtain a latent representation for predicting the user's preference value for an item.

Similar to RLBL, RIB introduces different behavior information into the input side of the RNN (in this case GRU), but the difference is that RIB uses an embedding matrix to represent multiple behavior information, where each behavior corresponds to an embedding vector. RIB also uses an attention layer to capture the importance of different behaviors, and in the original paper, the modeling of dwell time was also considered. Nevertheless, RIB may capture limited real user behavior information since it uses an embedding matrix to represent behavior types and then concatenates them directly with the item embedding.

BINN. BINN [28], based on LSTM, models users' long-term and short-term preferences to improve the next-item recommendation performance. BINN takes a sequence of (item, behavior) pairs as input and models each sequence from a local perspective. BINN contains two modules: session behaviors learning (SBL) to model a user's current consumption motivation, and preference behaviors learning (PBL) to learn the user's historical stable preference. In SBL, a context-aware LSTM (CLSTM) incorporating the behavior information as input is built, whose input gate $\boldsymbol{i}_t$, forgetting gate $\boldsymbol{f}_t$, output gate $\boldsymbol{o}_t$, internal state $\boldsymbol{c}_t$, and external state $\boldsymbol{h}_t$ at the tth time step are as follows:

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{\mathrm{vi}} V_{i_u^t} + \boldsymbol{W}_{\mathrm{hi}} \boldsymbol{h}_{t-1} + \boldsymbol{W}_{\mathrm{ci}} \boldsymbol{c}_{t-1} + \boldsymbol{W}_{\mathrm{bi}} F_{f_u^t} + \boldsymbol{b}_{\mathrm{i}}), \tag{17}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{\mathrm{vf}} V_{i_u^t} + \boldsymbol{W}_{\mathrm{hf}} \boldsymbol{h}_{t-1} + \boldsymbol{W}_{\mathrm{cf}} \boldsymbol{c}_{t-1} + \boldsymbol{W}_{\mathrm{bf}} F_{f_u^t} + \boldsymbol{b}_{\mathrm{f}}), \tag{18}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \tanh(\boldsymbol{W}_{\mathrm{vc}} V_{i_u^t} + \boldsymbol{W}_{\mathrm{hc}} \boldsymbol{h}_{t-1} + \boldsymbol{W}_{\mathrm{bc}} F_{f_u^t} + \boldsymbol{b}_{\mathrm{c}}), \tag{19}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{\mathrm{vo}} V_{i_u^t} + \boldsymbol{W}_{\mathrm{ho}} \boldsymbol{h}_{t-1} + \boldsymbol{W}_{mathrmco} \boldsymbol{c}_t + \boldsymbol{W}_{\mathrm{bo}} F_{f_u^t} + \boldsymbol{b}_{\mathrm{o}}), \tag{20}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \tanh(\boldsymbol{c}_t), \tag{21}$$

where $\boldsymbol{W}_{(\cdot)} \in \mathbb{R}^{d\times d}$ are the internal model parameters of the LSTM. Then the output $\boldsymbol{h}_t$ at the last time step t can be served as the user's current consumption motivation representation $\boldsymbol{h}_{\mathrm{SBL}}$. In PBL, BINN adopts a bidirectional CLSTM (Bi-CLSTM) which considers both forward and backward input sequences to obtain the long-term preference representation $\boldsymbol{h}_{\mathrm{PBL}}$. By concatenating $\boldsymbol{h}_{\mathrm{SBL}}$ with $\boldsymbol{h}_{\mathrm{PBL}}$, the obtained representation is utilized to make predictions and generate recommended items.

BINN proposes a novel gating structure, consisting of Bi-CLSTM and CLSTM, which enables the memorization of multi-behavior information in sequences. In contrast to RLBL and RIB in how to introduce multi-behavior information, BINN modifies the internal structure of the LSTM by feeding the behavior embedding matrix into the Bi-CLSTM and CLSTM to make it suitable for multi-behavior sequences. However, the limitations of BINN are similar to those of RIB, as both introduce behavior embedding matrices in the input layer only, which may result in the loss of latent behavior information and the inability to capture dependencies among behaviors.

IARS. IARS [34] is also a work that incorporates the item category to perform the next-item recommendation task. IARS consists of four blocks in total, which are an RNN-based encoder for perceiving user intent, and three decoders, i.e., a judgment or prediction task based on user intent, so as to learn the complex and co-existing intent of the user. Specifically, the encoder takes a sequence of (item, behavior, category) tuples as input, adopts a local modeling perspective, and processes the behavior types through multiple multi-behavior GRU units (MGRUs) to capture multiple intentions of the user. Note that we only discuss Soft-MGRU, one type of MGRU, for its lower spatial complexity and better performance by sharing the same set of parameters among different behaviors. After an embedding layer, the item embedding, category embedding, and behavior embedding are fed into Soft-MGRU to obtain the hidden state $\boldsymbol{h}_t$ at the time step t.

Soft-MGRU encodes the dependencies of items in multi-behavior sequences and obtains hidden states that characterize user intentions. It takes into account item categories and utilizes an attention network to capture the user's purchase intention for candidate items. The introduction of multi-behavior information is also achieved through the GRU's input, which concatenates the behavior embedding, the item embedding, and the category embedding. However, the behavior embedding only participates in the computation of the reset gate and update gate, which only addresses the challenge of sequence modeling for heterogeneous behavioral feedback, failing to capture the global behavior dynamics and the relationships among different behaviors.

MBN. MBN [44] models multi-behavior sequences towards the next-basket recommendation problem. The MBN architecture is composed of three modules, i.e., basket encoder, meta multi-behavior sequence encoder, and recurring-item-aware predictor. Specifically, the basket encoder converts the item representation $\boldsymbol{e}_v$ to the basket representation of the items $\boldsymbol{E}_{u,t}^f$ by a max pooling method. In the meta multi-behavior sequence encoder, multiple behavior-specific subsequences of items are taken as input and go through Behavior-RNN layers to learn behavior-specific information, which is local in the perspective of modeling. In addition to the Behavior-RNN layers, this work also proposes a Meta-RNN layer to learn the collective knowledge of multi-behavior sequences. Then, a gathering-scattering scheme is utilized to correlate the Meta-RNN layer and the Behavior-RNN layer. The representations learned by the Behavior-RNN layer are gathered to the Meta-RNN layer to learn the collective knowledge of multi-behavior sequences, and then the representations learned by the Meta-RNN layer are scattered to the individual Behavior-RNN layers to calibrate behavior modeling. In the recurring-item-aware predictor, a mixed probabilistic function in the generate mode and the repeat mode is proposed to predict the probability of each item in the next basket, which can simulate the distribution of items with biased repetition.

MBN introduces a method of gathering and then scattering to fuse and assign the learned multi-behavior information to different Behavior-RNN layers at the Meta RNN layer, which is a more explicit way to model intra-behavior and inter-behavior sequential information. In addition, any type of user behavior can be treated as the target behavior type. Nonetheless, as the number of behavior types increases, the number of behavioral RNN layers and associated parameters also increases, resulting in heightened computational complexity. Moreover, the division of the item basket in MBN is based on the time span, which may not align with the real-world scenario of purchasing a basket of items at the same time.

In addition to the above studies, several efforts employ RNN-based learning architectures to model the sequentiality and heterogeneity of user behaviors. CBS [82] models longer sequences rather than short-term dependencies for the next-basket recommendation problem with the use of an LSTM with or without shared parameters for each of the two behaviors (or the representation obtained from the embedding layer directly for the target behavior sequence). AIR [80] introduces item categories and proposes an attentional RNN to model the user's intention transitions. DIPN [83] employs a GRU and a hierarchical attention mechanism to effectively capture heterogeneous user behaviors and utilizes a multi-task module to capture short-term and long-term purchase preferences. HUP [33] utilizes the attention mechanism, and designs LSTMs with the addition of behavior gate and time gate at the micro-, item-, and category-levels to capture different granularities of information from session-based recommendation. In terms of federated recommendation, DeepRec [84] applies a GRU on the historical interaction data of all users on the cloud. The model is then pushed to users' devices, which makes it possible to fine-tune it for individuals to obtain a personal recommendation model for each of them. MAINT [81] models the input interaction sequences through a novel LSTM, which adds behavior specifics in the gates, and utilizes an attention mechanism to extract users' multifaceted preferences guided by the target behavior type (i.e., purchases) sequence, as well as adaptively fuses these intentions through a gating mechanism. DyMuS [35] encodes each behavior-specific subsequence of items by dynamic GRUs and incorporates the obtained information by dynamic routing at the sequence level to capture the user's dynamic preferences, and its improved version DyMus+ leverages dynamic routing again to encode each behavior subsequence at the item level to capture the correlation between items within the behavior-specific subsequences.

In summary, the RNN-based learning architecture is suitable for sequence problems and can store short-term memories, but suffers from the gradient disappearance and gradient explosion problems. In addition, RNN is inefficient and has difficulty in predicting information about future sequences since the output of the current moment depends on the computation and the output of the previous moment. At present, the industry rarely leverages RNN-based learning architecture for recommendations.

4.2 GNN-based learning architecture

GNNs [85, 86], which are utilized to extract features, are a widespread technique in recent years, and there have been many excellent graph neural network models, including GCN [87], GraphSAGE [88], and GAT [89]. They can fully exploit the higher-order neighbor information of nodes and perform well on recommender systems.

4.2.1 *Basic paradigm*

In general, graph neural network models use graph convolution to allow nodes to obtain information about their neighbors. To make the procedure more specific, an example is shown in Figure 5(b), which depicts four nodes, labeled as nodes 1, 2, 3, and 4. Node 1's first-order neighbors are nodes 2 and 3. During the first-order graph convolution, the embeddings of nodes 2 and 3 are aggregated into the embedding of node 1. In the second-order graph convolution, node 3 is a neighbor of node 4. Since node 3 has already obtained information about node 4 in the first-order graph convolution, node 1 is able to obtain information about its second-order neighbor, node 4, during the second-order graph convolution. This allows the graph convolutional network to effectively utilize information from higher-order neighbors of nodes.

Table 3 Data & modeling perspectives and features used in studies based on GNN learning architecture.

Studies	Data perspective	Model perspective	Features
MGNN-SPred [30]	Some behavior-specific subsequences of items	Global	Modeling behaviors from behavior transition relations, containing homogeneous behavior transitions intra each kind of behavior-specific subsequences
DMBGN [90]	Some behavior-specific subsequences of items	Global	Focus on the task of voucher redemption rate prediction and model the relationship between multiple behaviors and vouchers effectively
GNNH [91]	Some behavior-specific subsequences of items	Local + Global	Capture both representations in sequential and non-sequential patterns to capture dynamic interests in a session, introduce the category attribute, leverage attention mechanism
GPG4HSR [38]	A sequence of (item, behavior) pairs	Local + Global	Learn various behavior transition relations from the global graph and the personalized graph, respectively
BGNN [39]	Some behavior-specific subsequences of items	Global	Construct directed graphs for different behavior transition (homogeneous and heterogeneous) information
BA-GNN [92]	Some behavior-specific subsequence of items	Global	Construct directed graphs for different behavior-specific sequences, respectively
GHTID [93]	A sequence of (item, behavior) pairs	Local + Global	Construct both global and local item-item graphs, mitigate the noise caused by auxiliary behaviors, leverage attention mechanism

4.2.2 *Methods in MBSR*

In MBSR, there are lots of studies achieving considerable recommendation performance based on GNN, such as MGNN-SPred [30], deep multi-behavior graph networks (DMBGNs) [90], GNNH [91], global and personalized graphs for heterogeneous sequential recommendation (GPG4HSR) [38], behavior-aware graph neural network (BGNN) [39], BA-GNN [92], and GHTID [93]. We describe some of them in detail below, and summarize the data perspective, the modeling perspective, and the characteristics of these studies in Table 3.

MGNN-SPred. MGNN-SPred [30] also utilizes GNN to model multi-behavior sequences in session-based recommendation scenarios from a global modeling perspective.

First, MGNN-SPred treats the multi-behavior sequences $\mathcal{S}_\mathrm{p}$ and $\mathcal{S}_\mathrm{e}$ as some sequences of behavior-specific items, and constructs a global graph from all the training sequences, where the nodes represent items, and the edges have two attributes, namely, purchase edges and examination edges. For example, the purchase edge of item a and item b means a user purchases item a and then purchases item b. For each node v, we can obtain four types of neighbor node subsets, i.e., $\mathcal{N}_{\mathrm{p}+}(v)$, $\mathcal{N}_{\mathrm{e}+}(v)$, $\mathcal{N}_{\mathrm{p}-}(v)$, and $\mathcal{N}_{\mathrm{e}+}(v)$. For example, $\mathcal{N}_{\mathrm{e}+}(v)$ and $\mathcal{N}_{\mathrm{e}-}(v)$ denote the incoming edges and outgoing edges of the node, which is treated as an examined item, respectively. The concrete forms of the neighbor node subsets of the node v are as follows:

$$\mathcal{N}_{\mathrm{p}+}(v) = \{v' \mid (v' \rightarrow v, \mathrm{purchase}) \in \mathcal{E}\}, \tag{22}$$
$$\mathcal{N}_{\mathrm{e}+}(v) = \{v' \mid (v' \rightarrow v, \mathrm{examination}) \in \mathcal{E}\}, \tag{23}$$
$$\mathcal{N}_{\mathrm{p}-}(v) = \{v' \mid (v \rightarrow v', \mathrm{purchase}) \in \mathcal{E}\}, \tag{24}$$
$$\mathcal{N}_{\mathrm{e}-}(v) = \{v' \mid (v \rightarrow v', \mathrm{examination}) \in \mathcal{E}\}, \tag{25}$$

where $\mathcal{E}$ denotes the edge set. Second, for a target item v, the k-level aggregated representations of four different neighbors (taking $\mathcal{N}_{\mathrm{p}+}(v)$ as an example) and the node representation $\boldsymbol{h}_v^k$ obtained from the final iteration can be calculated as follows:

$$\boldsymbol{h}_{\mathrm{p}+,v}^k = \frac{\sum_{v' \in \mathcal{N}_{\mathrm{p}+}(v)} \boldsymbol{h}_{v'}^{k-1}}{|\mathcal{N}_{\mathrm{p}+}(v)|}, \tag{26}$$
$$\boldsymbol{h}_v^k = \boldsymbol{h}_v^{k-1} + \boldsymbol{h}_{\mathrm{p}+,v}^k + \boldsymbol{h}_{\mathrm{e}+,v}^k + \boldsymbol{h}_{\mathrm{p}-,v}^k + \boldsymbol{h}_{\mathrm{e}-,v}^k, \tag{27}$$

where $\boldsymbol{h}_v^k \in \mathbb{R}^{d\times 1}$ and $\boldsymbol{h}_v^K \in \mathbb{R}^{d\times 1}$ denote the kth step and the last step of the item v representation in GNN, respectively. $\boldsymbol{h}_v^K$ is used as the corresponding item potential representation. Third, it treats the sequences as some behavior-specific subsequences of items and obtains the user's examination preference and purchase preference by aggregating all item potential representations of the examination sequence and the purchase sequence, respectively. The final preference representation is obtained after feeding the above two preferences to a fully connected layer and a gated network.

MGNN-SPred is a simple but effective method for GNNs to be directly applied to MBSR. By constructing a graph, the sequential occurrence relationships of different behaviors are reflected in the graph, allowing aggregation among different behaviors and enhancing their informational capability. Moreover, as the first model distinguishing behavior sequences individually before passing them through the gated neural network, MGNN-SPred effectively captures both intra-behavior and inter-behavior information, ensuring a well-balanced information representation. Due to these advantages of MGNN-SPred, some of the subsequent MBSR studies make improvements based on it [39, 40, 92]. For example, the improvement of BA-GNN [92] over MGNN-SPred is that BA-GNN constructs separate graphs for different behavior sequences, and utilizes GG-NNs [94] and a sparse self-attention mechanism to address the noise effect in the examination sequence, thus better capturing the information in multi-behavior sequences.

DMBGN. DMBGNs [90] focus on the task of voucher redemption rate prediction in the session-based recommendation scenario. It utilizes GNNs to model users' long-term voucher redemption preferences from a global perspective.

First, it treats the multi-behavior sequences $\mathcal{S}_{\text{atc}}$ and $\mathcal{S}_{\text{ord}}$ as some sequences of behavior-specific items, and divides them into four parts, i.e., $\mathcal{S}^{+}_{\text{atc}}$, $\mathcal{S}^{-}_{\text{atc}}$, $\mathcal{S}^{+}_{\text{ord}}$, and $\mathcal{S}^{-}_{\text{ord}}$. For example, the sequence $\mathcal{S}^{+}_{\text{atc}}$ means that the behaviors of add-to-cart happen before the behavior on the voucher, while the sequence $\mathcal{S}^{-}_{\text{atc}}$ means that the behaviors of add-to-cart happen after the behavior on the voucher. $\mathcal{S}^{+}_{\text{atc}}$, $\mathcal{S}^{-}_{\text{atc}}$, $\mathcal{S}^{+}_{\text{ord}}$, and $\mathcal{S}^{-}_{\text{ord}}$ are connected to the central voucher node by the closest items from the temporal perspective. We obtain four sub-graphs in the end, i.e., atc_{+}, atc_{-}, ord_{+}, and ord_{-}. Second, the four sub-graphs constructed above are fed into GNN with the Weisfeiler-Leman algorithm [95], separately. The representations of the four sequences $\mathcal{S}^{+}_{\text{atc}}$, $\mathcal{S}^{-}_{\text{atc}}$, $\mathcal{S}^{+}_{\text{ord}}$, and $\mathcal{S}^{-}_{\text{ord}}$ are concatenated and processed through MLP. Then, the final UVG embedding is generated by concatenating the output of the MLP function and the embedding of the central voucher node, and the score for the historical UVG is obtained by the dot-product between the two representations. Finally, the representation of the embedding calculated from the target UVG component is enhanced via an attention network.

GNN can model the relationship between multiple behaviors and vouchers effectively. In building the graph, it is also reasonable that the coupon node is only connected to nodes of other behaviors that are temporally close, which improves the relationship between temporally close nodes and the coupon. Furthermore, DMBGN incorporates all historical sequences into the GNN network, thus proficiently capturing users' long-term preferences. The output of these past sequences is subjected to attention, together with the output of the current sequence, effectively enhancing the information representation of the current sequence.

GPG4HSR. GPG4HSR [38] simultaneously considers the transition relationships between different behaviors and local contextual information, thereby improving the next-item recommendation performance. GPG4SHR focuses on two types of behaviors, i.e., examinations and purchases, and takes a sequence of (item, behavior) pairs as input to model all sequences and each sequence from a global and local perspective, respectively. Specifically, GPG4HSR first feeds the input into an embedding layer to obtain the item embedding $\boldsymbol{v}_{i^t}$, the behavior embedding F_{f^t} and the position embedding $\boldsymbol{p}_t$ of the item i^t interacted by the behavior type f^t at the time step t. Then the embeddings are introduced to a global graph layer and a personalization layer to capture the transition patterns between behaviors and users' intent considering adjacent contextual information, respectively. In the global graph layer, the input is the global node v_{i^t} of each item of a sequence (abbreviated as v) and all the edges linked to it in the global graph, where there are six edge types to distinguish specific behavior transitions, (i.e., e2e, p2p, e2p+, e2p−, p2e+, and p2e−) that considers the transition directions (inward or outward) between different behaviors on top of TransRec++. The corresponding neighbor node subsets are $\mathcal{N}_{\text{e2e}}(v)$, $\mathcal{N}_{\text{p2p}}(v)$, $\mathcal{N}_{\text{e2p+}}(v)$, $\mathcal{N}_{\text{e2p}-}(v)$, $\mathcal{N}_{\text{p2e+}}(v)$, and $\mathcal{N}_{\text{p2e}-}(v)$. The final generated neighbor group and behavior transition-specific representation are as follows (take e2p+ as an example):

$$\mathcal{N}_{\text{e2p+}}(v) = \{(v', \text{freq}) \mid (v' \to v, \text{freq}, \text{e2p+}) \in \mathcal{E}_{\text{g}}\}, \tag{28}$$

$$\boldsymbol{h}_v^{\text{e2p+}} = \frac{\sum_{(v',\text{freq}) \in \mathcal{N}_{\text{e2p+}}(v)} \text{freq} \times \boldsymbol{v}_{v'}}{\sum_{(v',\text{freq}) \in \mathcal{N}_{\text{e2p+}}(v)} \text{freq}}, \tag{29}$$

where $\boldsymbol{v}_{v'}$ is a concise representation of the node v' linked to the node v, freq $\in \mathbb{R}$ represents the frequency of the corresponding edge. Then the global graph representation $\boldsymbol{h}_v^g$ of the node v can be represented as the sum of the node representation with the weighted representation of all behavior transition representations.

In the personalization graph layer, the input contains the item embedding and the behavior embedding of the node v, i.e., $(\boldsymbol{v}_v + F_v)$, as well as the inward and outward attributes of node-connected edges, which can learn the importance of different behaviors and the adjacent context information, thereby capturing users' intent. The final graph representation $\boldsymbol{h}_v$ of the node v obtained by fusing the global graph representation $\boldsymbol{h}_v^g$ and the personalized graph representation $\boldsymbol{h}_v^u$:

$$\boldsymbol{h}_v^u = \boldsymbol{v}_v + F_v + \boldsymbol{h}_v^{+} + \boldsymbol{h}_v^{-}, \tag{30}$$

$$\boldsymbol{h}_v = \gamma_u \boldsymbol{h}_v^g + (1 - \gamma_u)\, \boldsymbol{h}_v^u, \tag{31}$$

where $\gamma_u = \sigma\left(\boldsymbol{W}_{gp}\left[\boldsymbol{h}_v^g; \boldsymbol{h}_v^u\right]\right)$. The final graph representation of the sequence $\boldsymbol{h}_u$ can be obtained by concatenating the graph representation of the corresponding nodes of the sequence, and then passed into a dropout layer and stacked self-attention blocks same as SASRec [15], together with the corresponding position embedding. The obtained representation is concatenated with the target behavior vector and fed into a softmax function to obtain the user's predicted preference value for the items.

GPG4HSR constructs both a global graph and a personalized graph, where the global graph is used to capture the relationships among heterogeneous behaviors, and the personalized graph is used to enhance the contextual representation of a single user's multi-behavior sequence for a better comprehension of the user's preferences. In addition, the graph construction has the time complexity $O(|\mathcal{R}|)$, where $\mathcal{R}$ denotes the set of user-item interactions, which makes it more efficient. Nevertheless, as the number of behavior types increases, the number of behavior transition relationship types also increases, which increases the complexity of graph construction. Moreover, the multi-order behavior transition relationships, which typically optimize performance, can pose challenges in modeling.

BGNN. BGNN [39] distinguishes between two different behavior sequences by utilizing a dual-channel learning strategy for the session-based recommendation. BGNN takes an examination sequence and a purchase sequence as input, and models the heterogeneous behavior transitions to obtain the semantic connections between diverse behaviors by two global graphs, i.e., homogeneous behavior transition graph (HoBTG) and heterogeneous behavior transition graph (HeBTG), so as to improve the recommendation performance.

Specifically, BGNN sends the examination sequence and the purchase sequence into the auxiliary channel and the target channel, respectively. In the target channel, the item representation is learned in the purchase sequence through HoBTG, which is basically equivalent to the modeling of the behavior transition relationship of MGNN-SPred. The auxiliary channel consists of three modules to learn the item presentation of the examination sequence. The first module directly uses homogeneous behavior transition in the target channel to obtain potential representation; the second module adaptively adjusts the contributions of different neighbors of nodes through an attention mechanism to learn the purchase-oriented item representation; and the third module is for representation aggregation, which is the representation of items obtained by balancing the above two modules by gathering. After obtaining the item presentation matrices of the examination sequence and the purchase sequence, the matrices are sent to an attention network separately and then fused together. Finally, the user's preference value for the item is obtained through a prediction layer.

BGNN constructs graphs that explicitly capture two behavior transition patterns of homogeneous and heterogeneous ones, and utilizes these graphs in the auxiliary behaviors to capture the contribution of the auxiliary behaviors to the target behaviors, thereby improving the user's next preferred item prediction under the target behaviors, though its training time is about 1.4 times that of MGNN-SPred. However, BGNN encounters difficulties in the setting of more behavior types, i.e., the transition relationships between behaviors become more complex with an increase in the number of behavior types, and additional graphs may also need to be constructed, thus increasing the complexity of the algorithm.

Apart from the above studies, there are some other studies utilizing GNN to address the MBSR problem. GNNH [91] treats behavior types and categories as features, constructs multi-relational item graphs and feature graphs from a global view, and further learns and fuses item and feature representations through GNN and attention mechanism from a local view for session-based recommendation. GHTID [93] constructs a local item-item transition graph and a global item-item co-occurrence graph, utilizing GCN to learn heterogeneous item transitions, and learns long-term and short-term interests under target behaviors through the attention mechanisms, thus mitigating the noise caused by auxiliary behaviors.

In summary, by constructing the user-item graph, GNN can be easily applied to the recommendation methods. For each graph node, the aggregation of neighbors' information allows each behavior to obtain information about other behaviors that occurred close in time. The recommendation performance is obviously enhanced by the information enhancement of the neighbor nodes. In comparison to RNN, GNN has the ability to model more complex relationships within multi-behavior sequences and possesses stronger capabilities to handle data sparsity.

4.3 Transformer-based learning architecture

The Transformer model [96], a deep learning architecture that utilizes a self-attention network to reduce the computational complexity and thus enhance the training speed, has gained wide recognition in recent years for its superior performance in sequence-to-sequence modeling. This model has been utilized in a wide range of research areas, including natural language processing [97, 98], computer vision [99, 100], and recommender systems [29, 42, 56].

Table 4 Data & modeling perspectives and features used in studies based on Transformer learning architecture.

Works	Data perspective	Model perspective	Features
DMT [29]	Some behavior-specific subsequences of items	Local + Global	Use the target item as a query, consider implicit feedback bias by a bias deep neural network
DFN [56]	Some behavior-specific subsequences of items	Local + Global	Use the target item as a query, consider implicit negative feedback noise by an attention network
ASLI [101]	A sequence of (item, behavior) pairs	Local	The addition of category and behavior embeddings is modeled as an interaction representation, on which a corresponding loss function is defined
DUMN [57]	Some behavior-specific subsequences of items	Local	Consider implicit feedback noise, use a memory network to obtain the long-term user preference
FeedRec [31]	Some behavior-specific subsequences of items and a sequence of (item, behavior) pairs	Local + Global	Consider implicit feedback noise by an attention network, consider multiple patterns of the multi-behavior sequences
NextIP [41]	Some behavior-specific subsequences of items and a sequence of (item, behavior) pairs	Local + Global	Treat the problem as the item prediction task and the purchase prediction task, consider multiple patterns of the multi-behavior sequences
MB-STR [18]	A sequence of (item, behavior) pairs	Local	A novel positional encoding function to model multi-behavior sequence relationships
FLAG [42]	A behavior-agnostic sequence of items and a sequence of behaviors	Local + Global	Model user's local preference, local intention, and global preference simultaneously
ITE [102]	A sequence of (item, behavior) pairs	Local	Conduct implicit to explicit modeling by distinguishing behaviors as explicit and implicit based on the strength of user preferences expressed by the behaviors, introduce the category attribute
DMBIN [103]	Some behavior-specific subsequences of items	Local	Enhance the differences and consistency across behaviors by two contrastive learning tasks, i.e., multi-behavior contrast and multi-behavior alignment
PBAT [36]	A sequence of (item, behavior) pairs	Local	Utilize Gaussian distribution to define entities and relations in multi-behavior sequences
MBSRec [104]	A sequence of (item, behavior) pairs	Local	Design a simple, efficient, and effective attentive model, along with a weighted behavior-specific loss

4.3.1 *Basic paradigm*

The basic architecture of the Transformer model is depicted in Figure 5(c). It consists of two modules: the encoders and the decoders. In this discussion, we focus on the encoders. The most crucial component within the encoder is the multi-head self-attention component. This component comprises several self-attention subcomponents, which are widely used in recommendation models. We specifically examine the self-attention component by considering the representation of an examination sequence $\mathcal{S}_{\mathrm{e}}$. The calculation of the self-attention component is as follows:

$$\boldsymbol{Q}_i = \mathrm{Em}(\mathcal{S}_{\mathrm{e}})\boldsymbol{W}_i^Q, \quad \boldsymbol{K}_i = \mathrm{Em}(\mathcal{S}_{\mathrm{e}})\boldsymbol{W}_i^K, \quad \boldsymbol{V}_i = \mathrm{Em}(\mathcal{S}_{\mathrm{e}})\boldsymbol{W}_i^V, \tag{32}$$

$$\boldsymbol{head}_i = \mathrm{softmax}\left(\frac{\boldsymbol{Q}_i^\top \boldsymbol{K}_i}{\sqrt{d_t}}\right)\boldsymbol{V}_i, \tag{33}$$

$$\boldsymbol{F}_{\mathrm{e}} = \mathrm{concatenate}\left(\boldsymbol{head}_1, \ldots, \boldsymbol{head}_h\right)\boldsymbol{W}^O, \tag{34}$$

where $\mathrm{Em}(\mathcal{S}_{\mathrm{e}}) \in \mathbb{R}^{n_{\mathrm{e}}\times d}$, $\boldsymbol{W}_i^Q \in \mathbb{R}^{d\times d_t}$, $\boldsymbol{W}_i^K \in \mathbb{R}^{d\times d_t}$, $\boldsymbol{W}_i^V \in \mathbb{R}^{d\times d_t}$ and $\boldsymbol{W}^O \in \mathbb{R}^{hd_t\times d}$ are the projection matrices. n_{e} is the length of the sequence $\mathcal{S}_{\mathrm{e}}$, d_t is the dimension of $\boldsymbol{K}_i$, and $\boldsymbol{F}_{\mathrm{e}}$ is the output of the self-attention component.

4.3.2 *Methods in MBSR*

In MBSR, there are some studies obtaining great recommendation performance based on Transformer, including DMT [29], deep feedback network (DFN) [56], ASLI [101], DUMN [57], FeedRec [31], next-item prediction and purchase prediction (NeIP) [41], MB-STR [18], FLAG [42], ITE [102], DMBIN [103], PBAT [36], and MBSRec [104]. We describe some of them in detail below, and summarize the data perspective, the modeling perspective, and characteristics of these studies in Table 4.

DMT. Deep multifaceted Transformers (DMT) [29] utilizes a multi-gate mixture-of-experts (MMoE) approach, a multi-task learning technique, to enhance the performance of both CTR and click value rate (CVR) predictions. Furthermore, it employs the Transformer model to analyze multi-behavior sequences from a local modeling perspective.

First, it treats the multi-behavior sequences as some sequences of behavior-specific items and inputs those into the encoder of Transformer. Take the examination sequence as an example, and the formula is as follows:

$$\boldsymbol{F}_{\mathrm{e}} = \mathrm{Encoder}\left(\mathrm{PE}\left(\mathrm{Em}(\mathcal{S}_{\mathrm{e}})\right)\right), \tag{35}$$

$$\boldsymbol{E}_{\mathrm{e}} = \mathrm{Decoder}(\boldsymbol{F}_{\mathrm{e}}, \mathrm{PE}(V_i^{\mathrm{target}})), \tag{36}$$

where $\mathrm{PE}(\cdot)$ is the positional encoding function, and it explicitly represents the sinusoidal positional embedding or the learned positional embedding [15, 96] in DMT. V_i^{target} is the embedding of the item to be predicted and one of the inputs for the decoder of Transformer. Second, $\boldsymbol{E}_{\mathrm{e}}, \boldsymbol{E}_{\mathrm{a}}$, and $\boldsymbol{E}_{\mathrm{o}}$ are concatenated and flattened with the normalized dense features gathered from the recommender systems and the target item embedding V_i^{target}. Third, the multi-task training model MMoE is used to improve the performance of both CTR and CVR prediction. In particular, DMT considers bias in implicit feedback, such as position and neighboring bias, and utilizes a deep neural network with the ReLU function.

DMT uses a Transformer with unshared parameters to capture the relationships within each behavior and subsequently feeds the different behavior features into the MMoE module, lacking the explicit modeling of the relationships between the different behaviors. A bias deep neural network is proposed for modeling implicit feedback bias, which is a good modeling solution.

DFN. DFN [56], another work for CTR prediction in ads, models multi-behavior sequences utilizing Transformer from a local modeling perspective and three modules commonly used in industry, i.e., a wide component, an FM component, and a deep component. We can draw a comparison between DFN and DMT [29]. First, like DMT, DFN employs a Transformer architecture with unshared parameters to capture the relationships within each behavior. It treats multi-behavior sequences as a series of behavior-specific items and inputs them into a multi-head self-attention mechanism. Second, DFN also takes into account the implicit feedback noise. Unlike DMT, DFN leverages the attention mechanism to explore the relationship between different behaviors, which can be advantageous. Note that the implicit negative feedback, i.e., the unexamination sequence $\mathcal{S}_{\mathrm{n}}$, is abundant in real life but contains noise. As such, DFN uses implicit positive feedback $\boldsymbol{f}_{\mathrm{e}}$ and explicit negative feedback $\boldsymbol{f}_{\mathrm{d}}$ to denoise the implicit negative feedback by an attention network. The formula is as follows:

$$\boldsymbol{f}_{\mathrm{ne}} = \mathrm{attention}(\mathrm{Em}(\mathcal{S}_{\mathrm{n}}), \boldsymbol{f}_{\mathrm{e}}), \tag{37}$$

$$\boldsymbol{f}_{\mathrm{nd}} = \mathrm{attention}(\mathrm{Em}(\mathcal{S}_{\mathrm{n}}), \boldsymbol{f}_{\mathrm{d}}), \tag{38}$$

where $\boldsymbol{f}_{\mathrm{e}} \in \mathbb{R}^{1\times d}$ and $\boldsymbol{f}_{\mathrm{d}} \in \mathbb{R}^{1\times d}$ are the keys, and $\boldsymbol{f}_{\mathrm{ne}} \in \mathbb{R}^{1\times d}$ and $\boldsymbol{f}_{\mathrm{nd}} \in \mathbb{R}^{1\times d}$ are the outputs of the two attention networks, respectively. Finally, $\boldsymbol{f}_{\mathrm{e}}$, $\boldsymbol{f}_{\mathrm{d}}$, $\boldsymbol{f}_{\mathrm{n}}$, $\boldsymbol{f}_{\mathrm{nc}}$, and $\boldsymbol{f}_{\mathrm{nd}}$ are concatenated and fed in the three modules commonly used in industry mentioned above with other features, i.e., item features, user profiles, and recommendation contexts.

In addition to DFN, two other studies also denoise the implicit feedback by an attention network with the help of the explicit feedback, the first of which, DUMN [57], also utilizes a memory network for modeling users' long-term preferences to perform the CTR prediction task, while the second work FeedRec [31], a work focusing on news recommendation, uses Transformers with shared and unshared parameters to perform user modeling.

NextIP. A dual-task learning approach towards the item prediction task and purchase prediction task (NextIP) [41] utilizes the self-attention mechanism to model multi-behavior sequences from a local modeling perspective and performs the next-item recommendation task. Unlike other methods, NextIP simultaneously treats the multi-behavior sequences as some sequences of behavior-specific items and a sequence of (item, behavior) pairs. Specifically, NextIP treats the multi-behavior sequential recommendation problem as two tasks, i.e., the item prediction task and the purchase prediction task.

In the item prediction task, the embeddings of behavior-specific and behavior-aware item sequences are entered into the self-attention block (SAB). Subsequently, NextIP proposes the target-behavior-aware context aggregator (TBCG) to fully model the interplay of different behaviors at different times. Specifically, TBCG takes the representations of the most recent interaction for behavior-specific subsequences as keys and values, takes the user's target behavior embedding as a query, and inputs those into the attention module and mean pooling function with the target behavior representations from the behavior-specific subsequence representations. Finally, the item prediction result is calculated by the inner product between the target item embedding and the representation added by the output of TBCG and the most recent interaction representation of the behavior-aware sequences.

In the purchase prediction task, the user's behavior sequence embeddings are input into the behavior-aware self-attention block, masked depending on user behavior types and behavior distance. Each auxiliary behavior representation from the output of the behavior-aware self-attention block is treated as a negative sample to model the user purchase preference.

In summary, NextIP proposes a new perspective on this multi-behavior sequential recommendation problem by framing it as both an item prediction and a purchase prediction task. This new perspective offers a fresh outlook on the issue at hand, allowing for more accurate and efficient solutions. Moreover, NextIP considers multiple input patterns of the multi-behavior sequences and uses the self-attention network to model multi-behavior sequences with good performance. The contrastive loss function used to train the model also contributes to recommendation performance.

MB-STR. MB-STR [18] utilizes a Transformer to model multi-behavior sequences from both global and local modeling perspectives, to address the next-item recommendation problem. MB-STR treats the multi-behavior sequence as a sequence of (item, behavior) pairs and feeds it into the multi-head self-attention network, which considers the sequential pattern and distinguishes it based on the types of behavior. Then, a parameter-shared network like MMoE is used to model the behavior-specific information, denoted as a behavior aware prediction (BA-Pred) module. BA-Pred includes two parts, i.e., the parameters-shared experts and the behavior-specific experts, where the latter are shared for the representations of the same behavior type.

In summary, MB-STR employs a range of behavior-specific parameters to represent diverse behavior sequences at a fine-grained level. This approach enables effective modeling of the distinctiveness and interdependence among various behaviors, rendering it a robust tool for behavior modeling. Meanwhile, the total number of parameters in MB-STR is $O(|\mathcal{V}|d + |\mathcal{B}|d^2 + n)$, and its time complexity is $O(n^2d + nd^2)$, which is moderate compared to other studies. Moreover, unlike the positional encoding function of the classical Transformer, MB-STR is inspired by T5 [105] in natural language processing and uses a novel positional encoding function to model multi-behavior sequence relationships, which can better capture their positional relationships.

FLAG. Feedback-aware local and global (FLAG) [42] takes into account both user intent and preference complexity in modeling multi-behavior sequences for next-item recommendation. It takes a behavior-agnostic sequence of items and a sequence of behaviors as input, and employs both the global and local modeling perspectives. FLAG has four parts, including a local preference modeling, a global preference modeling, a local intention modeling, and a prediction module.

In the local preference modeling, the input matrix $X_u^{(0)}$, composed of the element-wise additions of the item embedding and the position embedding, is fed into the multiple stacked feedback-aware self-attention blocks (FSABs), and then obtains a user's local preference $\boldsymbol{z}_t^{\mathrm{lp}}$ at time step t from the top FSAB. Specifically, an FSAB successively goes through a feedback-aware input layer with a mask mechanism, a self-attention layer, and a feed-forward layer. In the global preference modeling, the authors use a location-based attention layer to model users' global preferences $\boldsymbol{z}^{\mathrm{gp}}$. Given that the preferences of users, both local and global, cannot be effectively modeled through local preference modeling and global preference modeling alone, a feedback-based attention layer (FAL) is proposed for local intention modeling. It receives an input matrix $\boldsymbol{O}$ that takes into account both the examination-specific and purchase-specific embedding matrices:

$$\boldsymbol{o}_u^t = V_{i_u^t} + \boldsymbol{p}_t' + F_{f_u^t}, \tag{39}$$

$$\boldsymbol{O} = \left[\boldsymbol{o}_u^1; \ldots; \boldsymbol{o}_u^l; \ldots; \boldsymbol{o}_u^T\right], \tag{40}$$

where $V_{i_u^t} \in \mathbb{R}^{1\times d}, \boldsymbol{p}_t' \in \mathbb{R}^{1\times d}$, and $F_{f_u^t} \in \mathbb{R}^{1\times d}$ are the item-specific embedding vector, the position-specific embedding vector and the behavior-specific embedding vector f_u^t of the item i_u^t at time step t, respectively. The next behavior $F_{f_u^{t+1}}$ is treated as a query vector to uncover the user's local intention in the following time step, so as to obtain the final local intention feature $\boldsymbol{z}_t^{li}$. Then an item similarity gating (ISG) module is proposed to achieve a balance between the local and global preferences with a weight factor λ, and then the obtained balanced preference representation $\boldsymbol{z}_t^{\mathrm{lgp}}$ and the local intention feature $\boldsymbol{z}_t^{\mathrm{li}}$ are element-wise added to get the final representation $\boldsymbol{z}_t$ of the sequence at time step t.

FLAG models the user's local preference, global preference, and local intention with acceptable time complexity and space complexity, where the multiple behaviors are utilized as a mask matrix in the local preference learning module, and as part of the input to the module through behavior embedding for better distinguishing the user's different behaviors and consequently improving preference modeling. However, in the local intention learning module, FLAG uses the next real feedback as the query vector during training, which may have a data bias that allows the model to overfit the historical behavior data. Furthermore, this approach may not perform well in cold-start settings where there is little historical interaction data.

Apart from the above studies, there are some other Transformer-based models for MBSR. ASLI [101] models the item sequential information through the self-attention mechanism, and the interaction information with a hybrid of behavior types and categories through the depth-wise temporal convolutional networks. ITE [102] performs implicit-to-explicit modeling by classifying behaviors as explicit and implicit based on the strength of user preferences indicated by the behaviors, and employs the Bert architecture to extract users' long-term and short-term interests, and further enhances item representation and long-term interest learning with category information. DMBIN [103] encodes multi-behavior subsequences of items into corresponding behavior-specific interests and behavior-invariant interests based on an attention mechanism, and extracts the features of differences and consistency among behaviors through the two contrastive learning tasks. PBAT [36] represents the input sequences with a Gaussian distribution, where the entities include user, item, behavior, and position, and the relations conclude the behavior-relation. Then it combines the personality interest with personalized pattern learning through self-adaptive Gaussian production to better depict the user's personalized preferences. Then, PBAT integrates the unified behavior relations and the personalized patterns, applying the behavior-aware attention mechanism to explore the sequential collaborations from the item, behavior, and position perspectives, thus accurately exploring the user sequence dependencies. MBSRec [104] effectively captures multi-behavior sequential patterns using a simple self-attention mechanism and employs a weighted binary cross-entropy loss to precisely allocate different weights to different behavior types. MBSRec is both adaptable and effective, capable of scaling to an arbitrary number of behavior types while maintaining minimal effect on the training and inference overhead.

In summary, Transformer, a sequence-to-sequence model, has demonstrated exceptional performance in recommender systems. Typically, Transformer captures the temporal relationship of behaviors by incorporating positional information in MBSR. Through the utilization of an attention mechanism, it is able to model relationships both within and among behaviors. With superior parallel computing capabilities, an enhanced ability to capture long-term dependencies, and stronger interpretability, Transformer surpasses RNN and GNN in MBSR to some extent.

4.4 Generic-method-based learning architecture

Since there are a lot of relevant and advanced studies in a research area, it is necessary to study a generic framework that can utilize any of the previous relevant studies to obtain information. A learning architecture based on a generic method that can employ a particularly designed module on a state-of-the-art model, combined with some innovative modeling modules to enhance the performance of that model, is a direction worth further study.

For the MBSR problem, the most important issues to consider are how to model sequences and how to distinguish between different behaviors. As such, the use of generic-method-based learning architectures can be chosen to improve the recommendation performance by following previous effective models of SBSR or MBR in the modeling of sequences or heterogeneous behaviors. Behavior-aware recommendation (BAR) [106] is a generic framework utilized in terms of obtaining sequence representations, which we introduce below.

BAR. BAR proposes a generic learning architecture for modeling multi-behavior sequences from a global modeling perspective, including a behavior attention layer and a task-specific layer. In the behavior attention layer, an attention network is used to enhance the presentation of the item embedding. First, the embedding of an item ℓ is added by the behavior embedding $B_{b_u^\ell}$ and the position embedding P_ℓ. Then an attention network is used to obtain the attention score $\alpha_\ell \in \mathbb{R}$ representing the relationship between the behavior embedding $B_{b_u^\ell}$ and the new presentation of item embedding X_ℓ, and is added to the item embedding $V_{i_u^\ell}$ to learn the hidden representation at each time step:

$$\boldsymbol{h}_{t-1} = \mathrm{RM}((1+\alpha_{t-L})V_{i_u^{t-L}}, \ldots, (1+\alpha_{t-1})V_{i_u^{t-1}}), \tag{41}$$

where $\mathrm{RM}(\cdot)$ denotes some important components used in sequential recommendation methods, e.g., recurrent neural network and convolutional neural network. $\mathrm{RM}(\cdot)$ reflects the generality of BAR, as any SBSR method like SASRec [15] can be utilized as a module of $\mathrm{RM}(\cdot)$ to learn the potential representations of sequences.

The task-specific layer is proposed as a solution to address the challenge of the unknown status of whether the behavior is the purchase or not when the model is focused on predicting the next purchased item. It uses an MLP to obtain the connection between the sequential information representation $\boldsymbol{h}_{t-1}$ and the behavior embedding $B_{b_u^\ell}$.

In summary, a general framework such as BAR, which directly applies the modeling methods used in SBSR, possesses better performance and strong generalization capability, but now there are few studies aiming to enhance the performance of recommendations. Hence, it can be beneficial to investigate the generalizability of modeling behavior types and transitions or to propose a generic model that incorporates the items' knowledge graph and the social connections among users.

Table 5 Data & modeling perspectives and features used in studies based on hybrid learning architecture.

Studies	Hybrid techniques	Data perspective	Model perspective	Features
MKM-SR [17]	RNN + GNN	A behavior-agnostic sequence of items and a sequence of behaviors	Global	Consider the knowledge graph of the items and the attributes
MBGNN [40]	RNN + GNN	Some behavior-specific subsequences of items	Local + Global	Consider both behavior type and transition to distinguish different node sets
GBAN [107]	RNN + GNN	A sequence of (item, behavior) pairs	Local	The prediction is the probability of a user generating each behavior on the news; introduce item category, tag, and topic for interactive news recommendation
UMBGN [108]	RNN + GNN	A sequence of (item, behavior) pairs	Local + Global	Construct a user-item graph and utilize an attention mechanism to aggregate user-item interaction information
MMFSR [109]	CNN + GNN	A sequence of (item, behavior) pairs and some behavior-specific subsequences of items	Local + Global	Mitigate the user cold-start problem in session-based recommendation based on a meta-learning framework, leverage attention mechanism
MISD [110]	Transformer + CNN	A sequence of (item, behavior) pairs	Local	Capture the dynamic multiple interests of users, the first MBSR work to use the diffusion model for denoising, leverage the diffusion model and dynamic routing
KHGT [111]	Transformer + GNN	Some behavior-specific subsequences of items	Local + Global	Consider item-item relation information
MBHT [49]	Transformer + GNN	A sequence of (item, behavior) pairs	Local + Global	Model users' short-term and long-term preferences by self-attention network and graph neural network, respectively
TGT [50]	Transformer + GNN	A sequence of (item, behavior) pairs	Local + Global	Model long-term and short-term multi-behavior sequence features separately to model a user's dynamic preference
AMAN [112]	Transformer + GNN	Some behavior-specific subsequences of items	Local	Take the first interacted item in a session as the user's original interests
MMCLR [43]	Transformer + GNN	Some behavior-specific subsequences of items	Local + Global	Introduce three novel contrastive learning tasks
RCL [51]	Transformer + GNN	A sequence of (item, behavior) pairs	Local + Global	Capture the long and short-term interests of users and enhance user representations through contrastive learning
FHT-MB [113]	Transformer + GNN	A sequence of (item, behavior) pairs	Local	Capture the users' various periodic behavior patterns and dependencies across behavior types simultaneously
FATH [114]	Transformer + GNN	A sequence of (item, behavior) pairs	Local + Global	Learn local interaction and global dependencies through hypergraph networks and reduce memory usage and time cost through a flash attention

4.5 Hybrid-method-based learning architecture

Combining multiple technologies for modeling can make use of the advantages of different technologies, and different technologies can also complement each other, leading to the improvement of modeling ability. The effective integration of diverse technologies within different modules is a crucial aspect to be considered when utilizing a hybrid-method-based learning architecture.

MBSR needs to model the sequence and behavior types at the same time, and it also needs to consider long-term and short-term preferences, as well as local or global information, which provides opportunities for employing different technologies. In MBSR, there are some studies utilizing different techniques, including MKM-SR [17], MBGNN [40], GBAN [107], UMBGN [108], MMFSR [109], MISD [110], KHGT (knowledge-enhanced hierarchical graph Transformer network) [111], MBHT [49], TGT [50], AMAN [112], MMCLR [43], RCL [51], FHT-MB [113] and FATH [114]. We describe some of them in detail below, and summarize the data perspective, the modeling perspective, and the characteristics of these studies in Table 5.

MKM-SR. MKM-SR [17] utilizes the GG-NNs [115] and the GRU to model multi-behavior sequences from a global modeling perspective. Here, the global modeling perspective denotes that all sequences are modeled together, rather than each sequence separately. We focus on the part of MKM-SR that models user multi-behavior sequential information, i.e., M-SR. It treats the multi-behavior sequence as a sequence of items and a sequence of behaviors for modeling. Then M-SR utilizes GG-NNs and GRU to model the item sequence and the behavior sequence, respectively, and concatenates the output vectors to obtain the behavior characteristics of the user.

M-SR aggregates the embedding of the nodes by the constructed user-item graph. Subsequently, it is fed into the GRU module to enhance the information further. In comparison with the methods utilizing RNN alone, M-SR can capture the bidirectional sequence relationships. Furthermore, M-SR's methodology of framing the item sequence and inputting it into the GG-NNs proficiently models the relationships among all items. Additionally, utilizing the GRU to input behavior sequences, rather than GG-NNs, enables M-SR to effectively capture the user's behavior sequential preferences.

MMFSR. MMFSR [109] effectively mitigates the user cold-start problem in session-based recommendation based on a meta-learning framework. Specifically, MMFSR constructs a global item-item relationship graph and feeds it into the GNN along with multiple behavior-specific subsequences of items to obtain the representations under the auxiliary and target behaviors. Then, the obtained representations are input into a self-attention layer and a gate network successively for sequential encoding and integration. MMFSR also takes a sequence of (item, behavior) pairs as another part of the input, where a temporal convolutional network (TCN) is applied to capture the user's current intention. In the framework of meta-learning, MMFSR designs memory mechanisms to guide the initialization of parameters for each session by providing personalized biases, which include user intent memory, target behavior memory, and auxiliary behavior memory, thus enabling sessions with similar intentions to share relevant knowledge and mitigate the user cold-start problem in session-based recommendation.

MISD. MISD [110] proposes a dynamic multi-interest network and a simple diffusion approach to capture dynamic personalized interests and mitigate the noises present in implicit feedback, respectively, which is the first work that introduces the diffusion model [116] to MBSR. Specifically, in the dynamic multi-interest network, the input is a sequence of (item, behavior) pairs, which is then passed through a Transformer layer to obtain dual-scale behavior pattern representations. Then MISD explicitly models the obtained representations to obtain long- and short-term interests through dynamic routing, and then further distinguishes users' multiple interests through cross CNNs. In addition, MISD performs interest matching to dynamically match the most relevant interests with the current interaction features to mitigate the interest drift problem. In the simple diffusion model for denoising, MISD uses MLP as the approximator and does not corrupt the user interest representations to pure noise in the forward process to preserve personalization, which is similar to DiffRec [117]. Currently, the diffusion model has some relevant research in sequential recommendation [117–121], while the inherent noise present in multiple implicit feedback indicates the significant value of the diffusion model applied to multi-behavior sequence scenarios, which deserves further exploration.

KHGT. KHGT [111] also utilizes Transformer and graph neural network to model multi-behavior sequences from both global and local modeling perspectives, and treats the multi-behavior sequence as a sequence of behavior-specific item modeling. For the position information of the user multi-behavior sequences, KHGT designs a novel encoding position function, which takes into account the users, the items, and the behavior types. For the user-item graph, unlike other methods, it constructs a heterogeneous graph of all users and interacted items. Each edge represents a record of a user's interaction with an item under a certain behavior type. The item-item graph is constructed using the item relation information, such as the item category. To extract the transition information about the nodes, a behavior-specific multi-head self-attention network is employed, and then the information of the graphs is utilized to aggregate the neighborhood information of the learned node. Finally, the information of each node is obtained.

KHGT is one of the few approaches to incorporate an item-to-item relationship within MBSR. This integration effectively enhances the information pertaining to each item, resulting in improved recommendation performance. It constructs the user-item and item-item graphs and uses Transformer to model the relationship among different behaviors. The relationships within each behavior and between multiple behaviors are thoroughly considered and are thus modeled well.

Apart from the above studies, there are some MBSR methods utilizing a hybrid learning architecture. MBGNN [40] leverages GRU and GNN to model the user's global and local preferences, respectively, to solve the session-based recommendation problem, where behavior transition is considered in the construction of the graph similar to other GNN-based studies. GBAN [107] performs local modeling for each user based on a graph-based convolutional neural network and an attention-based LSTM to separately learn news representations and behavior sequence representations for interactive news recommendation. Unlike the mainstream approach in the industry of using CTR as a performance metric for binary classification tasks, the final output of GBAN is the probability of each behavior of the user on the candidate news. UMBGN [108] also models multi-behavior sequences based on GNN and GRU, while considering the local personality of a single user and the global association of all users. MBHT [49] and TGT [50] utilize GNN and Transformer to model users' long-term and short-term preferences, respectively, where the former designs a novel self-attention mechanism inspired by Linformer [122]. AMAN [112] captures item associations within and across behavior sequences based on GNN and Transformer, and adaptively captures co-dependencies between different behavior sequences for session-based recommendation. Unlike other MBSR studies, AMAN explicitly considers the user's original interest, i.e., the first item the user interacts with as his/her original interest, whereas this may have a discounted effect in long sequence scenarios. MMCLR [43] utilizes Bert4Rec [123] and lightGCN [124] to encode local sequential information and global graph information under each behavior type of users, and conducts three novel contrastive learning tasks. RCL [51] utilizes a multi-relational GNN and a dynamic cross-relational memory network based on the attention mechanism to capture users' short-term and long-term interests, respectively, while multi-behavior contrastive learning is utilized to enhance user representation. FHT-MB [113] designs a filter-enhanced multi-scale Transformer for capturing behavior-aware sequential patterns under item transitions with different periodic trends, and utilizes a hypergraph structure to learn multi-behavior dependencies. FATH [114] utilizes a hypergraph neural network to model higher-order user-item interactions, and leverages the flash attention [125] to optimize memory and training speed within the model.

In summary, the increasing use of hybrid-method-based learning architecture for studies in MBSR suggests that combining different techniques can leverage the strengths of these techniques and play a complementary role, thus enhancing the recommendation performance. Consequently, this is a direction worthy of further research.

4.6 Summary

There have been many studies exploring MBSR, and in addition to the study mentioned above, many other researchers have made attempts [126–136]. There are also some studies that introduce sequential information, but they focus more on multi-behavior modeling [137], or apply to specific scenarios such as social recommendation and short video recommendation [138–143], or are applied in multi-task recommendation or multi-scenario recommendation [144–146]. In conclusion, multi-behavior sequential recommendation has attracted a certain degree of attention from both academia and industry.

5 Experiments

To give a more intuitive picture of the difference in recommendation performance of different methods, we conduct experiments with some representative classical and recent MBSR methods on two real-world datasets. Specifically, we compared some models across the following branches. (i) Traditional matrix factorization-based method: TransRec++ [37]. (ii) RNN-based learning architecture: RLBL [16] and BINN [28]. (iii) GNN-based learning architecture: GPG4HSR [38]. (iv) Transformer-based models: NextIP [41] and FLAG [42]. (v) Generic-method-based learning architecture: BAR [106].

5.1 Data preparation

In the experiments, we use the following two real-world datasets: (i) UB[1)] is released by Aliyun Tianchi in 2021 and provided by Taobao, which consists of four behavior types, i.e., purchase, examination, add-to-cart and favorite; (ii) JD[2)] is collected from the competition of JD in 2019, which contains four behavior types, i.e., purchase, examination, comment and favorite. For both datasets, we regard purchase as the target behavior in the prediction phase.

The preprocessing steps for both UB and JD are as follows: (i) for duplicate (user, item, behavior) records, keep the one with the earliest time; (ii) remove the cold-start items that have been purchased fewer than n times, and the cold-start users that have purchased fewer than m times (here we set $n = 10$, $m = 5$ for UB, and $n = 20$, $m = 5$ for JD); (iii) sort the interaction records of each user according to the timestamp and divide them at the proportional position of 0.8 and 0.9, where the first 80% of the data is taken as the training set, 10% of the data between the cut-off points of 0.8 and 0.9 as the validation set, and the last 10% of the data as the test set; (iv) remove the user sequences where users have interactions in the validation set and the test set, but lack purchase history in the training set; (v) remove the cold-start items in the validation set and test set; (vi) in the training phase, regard the training set as training data and keep only the first purchased item of each user in the validation set as validation data. While in the test phase, merge all the interaction records in the validation set with those in the training set as the training data, and keep only the first purchased item of each user in the test set as the test data. We summarize the statistics of both datasets in Table 6.

1) https://tianchi.aliyun.com/dataset/dataDetail?dataId=42.
2) https://jdata.jd.com/html/detail.html?id=8.

Table 6 Statistics of the experimented datasets, where $\{p, v, c, r, f\}$ in the behavior types refer to purchase, examination, add-to-cart, comment and favorite, respectively.

Datasets	# Users	# Items	# Interactions	Avg. length	Density	Behavior types
UB	20,443	30,734	798,657	39.07	0.13%	$\{p, v, c, f\}$
JD	10,690	12,820	340,931	31.89	0.25%	$\{p, v, r, f\}$

5.2 Evaluation metrics

The two widely used metrics for top-K recommendation tasks [15, 49, 50, 111] are employed in our experiments, namely hit ratio (HR@K) and normalized discounted cumulative gain (NDCG@K), where the former focuses on the proportion of user-preferred items appearing in the recommendation list, and the latter focuses on the ordering of these items. The values of K are set to 5, 10, and 20. Note that larger HR@K and NDCG@K mean more recommendation performance gain. Following [147, 148], we perform the evaluation on the full list of candidate items.

5.3 Parameter settings

For a fair comparison, we set the batch size of all methods to 128, the embedding dimension to 50, and the maximum sequence length to 50 following [15, 41]. We also follow [15, 41] and adopt the Adam optimizer with a learning rate of 0.001, and set the dropout rate to 0.5. For the attention-based methods, we tune the number of blocks in $\{1, 2, 3\}$ and fix the number of heads to 1 due to the small dimension being unsuitable for subspace decomposition [15]. For other hyperparameters of some specific methods, we tune them according to the parameter ranges mentioned in the original papers. The hyperparameters are rigorously tuned on the validation data, and the optimal model is applied to predict the results on the test data. The reported experimental results represent the average values of three executions of the models.

5.4 Experiment results

We present our experimental results in Table 7, from which we have the following observations.

- The efficacy of deep learning methods. Traditional recommendation approaches, such as TransRec++, consistently underperform across all evaluation metrics, indicating that matrix factorization-based methods have inherent limitations in modeling user behaviors and boosting recommendation performance. In contrast, most deep learning approaches (e.g., BINN, NextIP, and FLAG) substantially enhance performance, thereby demonstrating the advantages of complex network architectures and multi-layer representations in multi-behavior sequential recommendation.
- The advantage of attention mechanisms in long-sequence tasks. The results in Table 7 show that Transformer-based methods excel at capturing long-term item dynamics. For example, FLAG achieves HR@20 and NDCG@20 scores of 0.0825 and 0.0438 on UB, respectively, significantly outperforming the other methods. This underscores the effectiveness of attention mechanisms in processing sequential data and capturing fine-grained contextual relationships.
- Performance difference among models within similar deep learning architectures. Even among models sharing the same deep learning framework, performance can vary considerably due to the differences in implementations and parameter settings. For instance, within the RNN-based architectures, BINN outperforms RLBL across all metrics on both datasets. This suggests that careful model design and task-specific optimization are essential to fully leverage the potential of a deep learning architecture.
- Challenges in model generalization. Some models (such as TransRec++, RLBL, NextIP, and BAR) exhibit inconsistent performance on UB, highlighting challenges in generalization when facing with varying sequence lengths and data densities, which suggests that model design should not solely focus on achieving high fitting accuracy on a given dataset but must also account for robust generalization across diverse application scenarios.

It is also important to note that these results are validated only under the current experimental settings. Under alternative conditions, for example, in scenarios with shorter sequences or lower data density, RNN and GNN models may exhibit more performance improvement [35, 39]. Moreover, hybrid methods do not necessarily outperform those relying exclusively on a single deep learning architecture, such as Transformer-based approaches [36, 104], further emphasizing the importance of selecting an appropriate model based on the specific task and setting.

Table 7 Recommendation performance on UB and JD. Note that bold scores stand for the highest scores among all methods, and underlined scores represent the next highest scores.

		MF	RNN		GNN	Transformer			Generic method
		TransRec++	RLBL	BINN	GPG4HSR	NextIP	FLAG	MBSRec	BAR
JD	HR@5	0.0447	0.0296	0.0398	0.0535	<u>0.0609</u>	**0.0622**	0.0540	0.0450
	NDCG@5	0.0309	0.0167	0.0220	0.0345	<u>0.0355</u>	**0.0364**	0.0318	0.0271
	HR@10	0.0833	0.0619	0.0889	0.0920	<u>0.1139</u>	**0.1241**	0.0956	0.0792
	NDCG@10	0.0432	0.0272	0.0377	0.0470	<u>0.0523</u>	**0.0563**	0.0452	0.0380
	HR@20	0.1195	0.1252	0.1542	0.1406	<u>0.1866</u>	**0.2010**	0.1627	0.1342
	NDCG@20	0.0522	0.0429	0.0543	0.0592	<u>0.0707</u>	**0.0758**	0.0621	0.0518
UB	HR@5	0.0142	0.0067	0.0239	0.0377	0.0361	**0.0457**	<u>0.0451</u>	0.0193
	NDCG@5	0.0095	0.0041	0.0170	0.0277	0.0285	<u>0.0333</u>	**0.0368**	0.0140
	HR@10	0.0204	0.0101	0.0337	0.0517	0.0469	**0.0622**	<u>0.0544</u>	0.0249
	NDCG@10	0.0115	0.0051	0.0202	0.0322	0.0319	<u>0.0387</u>	**0.0398**	0.0158
	HR@20	0.0287	0.0171	0.0495	<u>0.0717</u>	0.0571	**0.0825**	0.0673	0.0344
	NDCG@20	0.0136	0.0069	0.0241	0.0372	0.0345	**0.0438**	<u>0.0430</u>	0.0182

6 Future directions

The multi-behavior sequential recommendation (MBSR) problem, which is more representative of real-world recommendation scenarios, has increasingly gained attention from academia and industry in recent years. Although some studies with superior recommendation performance towards the MBSR problem have been proposed, there are still many issues worthy of further study. In this section, we discuss some potential future research directions for the MBSR problem, including data, techniques, optimization targets, large language models, trustworthiness, and responsibility.

Data. In the field of artificial intelligence, a comprehensive understanding of data is crucial for developing models. In the case of MBSR, the complexity of the data also poses various challenges when modeling. First of all, data sparsity has always been the focus of recommendation algorithms [149], and MBSR is no exception. However, excessive data sparsity can undermine the performance of association-based algorithms like collaborative filtering in recommender systems. Additionally, the multiple behaviors of MBSR make the pattern of data sparsity more intricate. In practical situations, such as cold-start settings, where new users or items are seldom interacted with, resolving the data sparsity issue is necessary to generate reasonable recommendations. Second, it is essential to explicitly model the data imbalance in MBSR. The data suffers from a heterogeneous behavior distribution problem similar to MBR and a sequence length problem similar to SBSR. User behavior distribution and interaction sequence lengths often differ in real-world scenarios. For instance, in shopping scenarios, users tend to make fewer purchases than examination behaviors, and users may examine varying item quantities. Third, there are several issues associated with data processing, including periodicity and noise. Periodicity refers to users' inclination to examine items at specific times, and noise refers to users examining items that do not align with their current preferences. While related studies have focused on denoising [56, 57], there remains a significant scope for further research, particularly in terms of how to explicitly model various types of specific noise, such as interactions that align with a user's long-term preferences but not their current preferences. As such, it is necessary to further explore how to deal with data sparsity, imbalance, periodicity, and noise, so as to improve the effectiveness of recommendations.

Techniques. Technical innovation has been the primary focus for most studies aiming to improve recommendation performance, yet several challenges remain with the current techniques used for MBSR. First, individual techniques have their own limitations. For example, Transformer can solve the problem of parallel computation that RNN is limited, but is less capable of capturing the local information than RNN due to the point-wise dot-product self-attention utilized [150, 151]. As such, combining multiple complementary components or techniques to solve the MBSR problem is an important research direction. Second, efficiency is an essential issue in MBSR due to the complexity of the data. It is worthwhile to investigate how to improve recommendation performance without sacrificing efficiency so as to enable real-time recommendations. Third, how to maintain acceptable time and space complexity when the number of behavior types increases is also a challenging issue. Fourthly, some studies propose models that perform well on some datasets but poorly on others during training and prediction [38, 42]. As such, it remains a challenge to improve the generalization of the models for MBSR. In addition, although there are some SR studies [20, 152, 153] utilizing data from different domains, or data with auxiliary information such as item category information, reviews, and knowledge graphs, it may be difficult to introduce such information into more complex MBSR scenarios. For instance, user behavior patterns may differ across different domains, and behaviors such as clicks that indicate users' weak preferences represent his or her unclear preferences for item attributes. As interactive conversational recommender systems become more prevalent between users and platforms, future MBSR techniques may need to model multi-behavior sequential data and multiple rounds of conversational text data. In summary, there is significant potential in the technical aspects of MBSR, especially in terms of combining methods, improving efficiency, and adapting to data diversity.

Optimization targets. Optimizing targets in MBSR also presents several challenges. Currently, most studies on MBSR focus on a single target, such as recommending more items that users would like to buy in a shopping scenario. However, the diversity of user behaviors allows the possibility of optimizing multiple targets simultaneously. For example, on the business side of the industry, there is often more than one single target to optimize [154], instead, there is a need to jointly optimize multiple targets, such as increasing the view rate and like rate of a video simultaneously. At present, the multi-target optimization methods mainly include setting sample weight, stacking multiple models, sharing model parameters for joint training, and MMoE [155]. However, there are some shortcomings in these methods. For example, in the method of stacking multiple models, the models are independent of each other, which makes the training process prone to the situation of over-fitting, while the sharing of experts in MMoE among all tasks may bring bias to some tasks. In addition, SR problems in cross-domain or multi-domain settings require recommendation performance gains of multiple domains simultaneously [156–158], without the exception of MBSR problems in the same settings. As such, how to optimize multiple objectives in a rational way is also a direction worth investigating.

Large language models. The remarkable performance of large language models (LLMs) [159,160] has received great attention within the academic community. A mounting body of research is presently dedicated to expansive language models in the domain of recommender systems [161,162]. Due to the large amount of textual information intrinsic to the recommendation task itself, as well as the commendable language comprehension ability and external knowledge reserve of the large model, modeling the representation of users and items with text information may hopefully supplant conventional ID-based paradigms. In the task of multi-behavior sequential recommendation, in addition to modeling sequence relationships, different behavior information also needs to be modeled for items. How to use a large model to effectuate a synergistic amalgamation of distinct behavior signals with the textual profiles of users and items is a relatively new direction. Note that items may have different relationships under different behaviors. To illustrate, if a user has examined a certain type of item, he or she may buy the same type of item from a different brand. However, after acquiring one of these items, the user may be less inclined to buy items from other brands. Therefore, it is necessary to integrate multi-behavior information into the modeling of large models and recommendation tasks, rather than simple sequence modeling.

Trustworthiness and responsibility. The need to build more trustworthy and responsible recommender systems has been raised when recommender systems consistently pursue higher accuracy, and are determined to recommend items to users transparently, fairly, and unbiasedly. Explainability and security are two main aspects of the trustworthiness of recommender systems [163], which also require further attention in MBSR. First, in terms of explainability, the complexity of the behaviors in MBSR makes the deep learning model less explainable. Attention is a common approach applied to MBSR to improve the explanation of deep learning models [32,49,56,106]. Second, in terms of security, the issue of privacy protection is becoming increasingly important to the state and to the public. Recommender systems need to avoid the problem of user information leakage when designing a model, including the risk of information leakage between users, between platforms, and between both users and platforms. For the MBSR problem, the user's behaviors are considered private. Such private-sensitive data can only be observed on the user's own client and cannot be uploaded to the cloud, thereby complicating the modeling process. To address this challenge, the federated recommendation, one of the most effective and popular approaches that address the privacy protection problem in recommender systems, was first proposed by Google in 2016 [164]. Not much work has been done to consider privacy and security in MBSR [84], and the use of federated learning [165] to secure privacy is an interesting direction. As such, it is important to build trustworthy and responsible recommender systems with higher explainability under the requirement of privacy protection, so that users can be fairly recommended the items they are interested in.

7 Conclusion

MBSR combines SBSR and MBR, requiring the modeling of both sequential information and heterogeneous behaviors, which provides some challenges while allowing for some optimization in recommendation performance. MBSR is closer to the range of user feedback that occurs in real-world scenarios. The growing body of MBSR studies in both academia and industry (though still fewer than that for SBSR) highlights the importance of MBSR in recommender systems. The increasing amount of studies for MBSR in academia and industry, although much fewer than those for SBSR, indicates the importance of MBSR in recommender systems. In this paper, we first introduce the MBSR problem in detail, followed by a classification of related studies, encompassing traditional methods and deep learning-based methods.

Due to the complexity of the MBSR problem, lots of studies use deep learning-based methods, including RNN, GNN, and Transformer, or their generic and hybrid architectures. For each of these learning architectures, we present their general form before transitioning to how to apply the learning architecture to the MBSR problem based on previous studies. For each work, we introduce it from the technology, data, and modeling perspectives, and discuss its strengths and weaknesses.

Through a detailed discussion of the studies that have been done so far, we find that MBSR still faces many challenges. In response to these challenges, we suggest five possible future directions, including data, techniques, security, optimization targets, and explanation, which we hope will give the readers some guidance on how to solve the MBSR problem better.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62461160311, 62172283, 62272315), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515010122), and National Key Research and Development Program of China (Grant No. 2023YFF0725100). We thank Mr. Jinwei LUO for his helpful discussions.

References

1 Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng, 2005, 17: 734–749
2 Ricci F, Rokach L, Shapira B. Recommender Systems Handbook. 2nd ed. New York: Springer, 2015
3 Gai L, Lei L. Dual collaborative topic modeling from implicit feedbacks. In: Proceedings of the 2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics, 2014. 395–404
4 Li G, Ou W. Pairwise probabilistic matrix factorization for implicit feedback collaborative filtering. Neurocomputing, 2016, 204: 17–25
5 Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, 2009. 452–461
6 Pan W, Zhong H, Xu C, et al. Adaptive Bayesian personalized ranking for heterogeneous implicit feedbacks. Knowl-Based Syst, 2015, 73: 173–180
7 Qiu H, Liu Y, Guo G, et al. BPRH: Bayesian personalized ranking for heterogeneous implicit feedback. Inform Sci, 2018, 453: 80–98
8 Tang L, Long B, Chen B, et al. An empirical study on recommendation with multiple types of feedback. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 283–292
9 Pan W, Liu M, Ming Z. Transfer learning for heterogeneous one-class collaborative filtering. IEEE Intell Syst, 2016, 31: 43–49
10 Pan W, Yang Q, Cai W, et al. Transfer to rank for heterogeneous one-class collaborative filtering. ACM Trans Inf Syst, 2019, 37: 1–20
11 Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, 2010. 811–820
12 Zeng Z, Lin J, Li L, et al. Next-item recommendation via collaborative filtering with bidirectional item similarity. ACM Trans Inf Syst, 2019, 38: 1–22
13 Tang J, Wang K. Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the 11th ACM International Conference on Web Search and Data Mining, 2018. 565–573
14 Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based recommendations with recurrent neural networks. In: Proceedings of the 4th International Conference on Learning Representations, 2016
15 Kang W C, McAuley J. Self-attentive sequential recommendation. In: Proceedings of the 2018 IEEE International Conference on Data Mining, 2018. 197–206
16 Liu Q, Wu S, Wang L. Multi-behavioral sequential prediction with recurrent log-bilinear model. IEEE Trans Knowl Data Eng, 2017, 29: 1254–1267
17 Meng W, Yang D, Xiao Y. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020. 1091–1100
18 Yuan E, Guo W, He Z, et al. Multi-behavior sequential transformer recommender. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022. 1642–1652
19 Fang H, Zhang D, Shu Y, et al. Deep learning for sequential recommendation. ACM Trans Inf Syst, 2020, 39: 1–42
20 Chen S, Xu Z, Pan W, et al. A survey on cross-domain sequential recommendation. ArXiv:2401.04971
21 Chen X, Lin L, Pan W, et al. A survey on heterogeneous one-class collaborative filtering. ACM Trans Inform Syst, 2020, 38: 1–54
22 Smith B, Linden G. Two decades of recommender systems at Amazon.com. IEEE Internet Comput, 2017, 21: 12–18
23 Wang J, Huang P, Zhao H, et al. Billion-scale commodity embedding for e-commerce recommendation in Alibaba. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018. 839–848

24 Covington P, Adams J, Sargin E. Deep neural networks for YouTube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, 2016. 191–198

25 Gomez-Uribe C A, Hunt N. The Netflix recommender system. ACM Trans Manage Inf Syst, 2015, 6: 1–19

26 Liu J, Dolan P, Pedersen E R. Personalized news recommendation based on click behavior. In: Proceedings of the 15th International Conference on Intelligent User Interfaces, 2010. 31–40

27 Wu F, Qiao Y, Chen J H, et al. Mind: a large-scale dataset for news recommendation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. 3597–3606

28 Li Z, Zhao H, Liu Q, et al. Learning from history and present: next-item recommendation via discriminatively exploiting user behaviors. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018. 1734–1743

29 Gu Y, Ding Z, Wang S, et al. Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, 2020. 2493–2500

30 Wang W, Zhang W, Liu S, et al. Beyond clicks: modeling multi-relational item graph for session-based target behavior prediction. In: Proceedings of the Web Conference 2020, 2020. 3056–3062

31 Wu C, Wu F, Qi T, et al. Feedrec: news feed recommendation with various user feedbacks. In: Proceedings of the ACM Web Conference 2022, 2022. 2088–2097

32 Zhou M, Ding Z, Tang J, et al. Micro behaviors: a new perspective in e-commerce recommender systems. In: Proceedings of the 11th ACM International Conference on Web Search and Data Mining, 2018. 727–735

33 Gu Y, Ding Z, Wang S, et al. Hierarchical user profiling for e-commerce recommender systems. In: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020. 223–231

34 Xu Y, Zhu Y, Yu J. Modeling multiple coexisting category-level intentions for next item recommendation. ACM Trans Inform Syst, 2021, 39: 1–24

35 Cho J, Hyun D, won Lim D, et al. Dynamic multi-behavior sequence modeling for next item recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 4199–4207

36 Su J, Chen C, Lin Z, et al. Personalized behavior-aware transformer for multi-behavior sequential recommendation. In: Proceedings of the 31st ACM International Conference on Multimedia, 2023. 6321–6331

37 Zhan Z X, He M K, Pan W K, et al. TransRec++: translation-based sequential recommendation with heterogeneous feedback. Front Comput Sci, 2022, 16: 162615

38 Chen W, He M, Ni Y, et al. Global and personalized graphs for heterogeneous sequential recommendation by learning behavior transitions and user intentions. In: Proceedings of the 16th ACM Conference on Recommender Systems, 2022. 268–277

39 Luo J, He M, Pan W, et al. Bgnn: behavior-aware graph neural network for heterogeneous session-based recommendation. Front Comput Sci, 2023, 17: 175336

40 Pan W, Yang K. Multi-behavior graph neural networks for session-based recommendation. In: Proceedings of the 3rd International Conference on Machine Learning, Big Data and Business Intelligence, 2021. 756–761

41 Luo J, He M, Lin X, et al. Dual-task learning for multi-behavior sequential recommendation. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management, 2022. 1379–1388

42 He M, Lin J, Luo J, et al. Flag: a feedback-aware local and global model for heterogeneous sequential recommendation. ACM Trans Intell Syst Tech, 2022, 14: 1–22

43 Wu Y, Xie R, Zhu Y, et al. Multi-view multi-behavior contrastive learning in recommendation. In: Proceedings of the International Conference on Database Systems for Advanced Applications, 2022. 166–182

44 Shen Y, Ou B, Li R. Mbn: towards multi-behavior sequence modeling for next basket recommendation. ACM Trans Knowl Discov Data, 2022, 16: 1–23

45 Koren Y. Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009. 447–456

46 Liu Q, Zeng X, Liu C, et al. Mining indecisiveness in customer behaviors. In: Proceedings of the 2015 IEEE International Conference on Data Mining, 2015. 281–290

47 Zhang F, Yuan N J, Lian D, et al. Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 353–362

48 Wu B, Mei T, Cheng W, et al. Unfolding temporal dynamics: predicting social media popularity using multi-scale temporal decomposition. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016. 272–278

49 Yang Y, Huang C, Xia L, et al. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In: Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2022. 2263–2274

50 Xia L, Huang C, Xu Y, et al. Multi-behavior sequential recommendation with temporal graph transformer. IEEE Trans Knowl Data Eng, 2022, 35: 6099–6112

51 Wei W, Xia L, Huang C. Multi-relational contrastive learning for recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems, 2023. 338–349

52 Liu N N, Xiang E W, Zhao M, et al. Unifying explicit and implicit feedback for collaborative filtering. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010. 1445–1448

53 Hadash G, Shalom O S, Osadchy R. Rank and rate: multi-task learning for recommender systems. In: Proceedings of the 12th ACM

Conference on Recommender Systems, 2018. 451–454
54 Zhao X, Zhang L, Ding Z, et al. Recommendations with negative feedback via pairwise deep reinforcement learning. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018. 1040–1048
55 He X, Zhang H, Kan M, et al. Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2016. 549–558
56 Xie R, Ling C, Wang Y, et al. Deep feedback network for recommendation. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020. 2519–2525
57 Bian Z, Zhou S, Fu H, et al. Denoising user-aware memory network for recommendation. In: Proceedings of the 15th ACM Conference on Recommender Systems, 2021. 400–410
58 Johnson C C. Logistic matrix factorization for implicit feedback data. In: Proceedings of Advances in Neural Information Processing Systems, 2014
59 Mnih A, Salakhutdinov R R. Probabilistic matrix factorization. In: Proceedings of Advances in Neural Information Processing Systems, 2007
60 Desrosiers C, Karypis G. A comprehensive survey of neighborhood-based recommendation methods. In: Recommender Systems Handbook. Boston: Springer, 2011. 107–144
61 Ma H, Yang H, Lyu M R, et al. Sorec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, 2008. 931–940
62 Mehta R, Rana K. A review on matrix factorization techniques in recommender systems. In: Proceedings of the 2nd International Conference on Communication Systems, Computing and IT Applications, 2017. 269–274
63 He R, Kang W, McAuley J. Translation-based recommendation. In: Proceedings of the 11th ACM Conference on Recommender Systems, 2017. 161–169
64 He R, McAuley J. Fusing similarity models with Markov chains for sparse sequential recommendation. In: Proceedings of the 16th IEEE International Conference on Data Mining, 2016. 191–200
65 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444
66 Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge: MIT Press, 2016
67 Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, 2008. 160–167
68 Socher R, Perelygin A, Wu J, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013. 1631–1642
69 Dahl G E, Yu D, Deng L, et al. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans Audio Speech Lang Process, 2011, 20: 30–42
70 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2012
71 Le Q V. Building high-level features using large scale unsupervised learning. In: Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013. 8595–8598
72 Gao C, He X, Gan D, et al. Learning to recommend with multiple cascading behaviors. IEEE Trans Knowl Data Eng, 2019, 33: 2588–2601
73 Zhou X, Liu D, Lian J, et al. Collaborative metric learning with memory network for multi-relational recommender systems. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019. 4454–4460
74 Wu C, Wu F, Cao Y, et al. Fedgnn: federated graph neural network for privacy-preserving recommendation. ArXiv:2102.04925
75 Wu C, Wu F, Lyu L, et al. Fedctr: federated native ad ctr prediction with cross platform user behavior data. ACM Trans Intell Syst Tech, 2022, 13: 1–19
76 Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model. In: Proceedings of the Interspeech, 2010. 1045–1048
77 Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw, 1994, 5: 157–166
78 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput, 1997, 9: 1735–1780
79 Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. ArXiv:1412.3555
80 Chen T, Yin H, Chen H, et al. Air: attentional intention-aware recommender systems. In: Proceedings of the 2019 IEEE 35th International Conference on Data Engineering, 2019. 304–315
81 Liu H, Ding J, Zhu Y, et al. Modeling multi-aspect preferences and intents for multi-behavioral sequential recommendation. Knowl-Based Syst, 2023, 280: 111013
82 Le D, Lauw H W, Fang Y. Modeling contemporaneous basket sequences with twin networks for next-item recommendation. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018. 3414–3420
83 Guo L, Hua L, Jia R, et al. Buying or browsing?: predicting real-time purchasing intent using attention-based deep network with multiple behavior. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019. 1984–1992
84 Han J, Ma Y, Mei Q, et al. Deeprec: on-device deep learning for privacy-preserving sequential recommendation in mobile commerce. In: Proceedings of The Web Conference 2021, 2021. 900–911

85 Scarselli F, Gori M, Ah Chung Tsoi M, et al. The graph neural network model. IEEE Trans Neural Netw, 2008, 20: 61–80

86 Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst, 2020, 32: 4–24

87 Welling M, Kipf T N. Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representation, 2016

88 Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017. 1025–1035

89 Velickovic P, Cucurull G, Casanova A, et al. Graph attention networks. In: Proceedings of the 6th International Conference on Learning Representation, 2018

90 Xiao F, Li L, Xu W, et al. DMBGN: deep multi-behavior graph networks for voucher redemption rate prediction. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2021. 3786–3794

91 Yu B, Zhang R, Chen W, et al. Graph neural network based model for multi-behavior session-based recommendation. Geo Inform, 2022, 26: 429–447

92 Liang Y, Song Q, Zhao Z, et al. BA-GNN: behavior-aware graph neural network for session-based recommendation. Front Comput Sci, 2022, 6: 1–16

93 Li X, Chen H, Yu J, et al. Global heterogeneous graph and target interest denoising for multi-behavior sequential recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, 2024. 387–395

94 Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks. In: Proceedings of the 4th International Conference on Learning Representations, 2016

95 Morris C, Ritzert M, Fey M, et al. Weisfeiler and Leman go neural: higher-order graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 4602–4609

96 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems, 2017

97 Devlin J, Chang M W, Lee K, et al. Bert: pre-training of deep bidirectional transformers for language understanding. ArXiv:1810.04805

98 Liu Y, Ott M, Goyal N, et al. Roberta: a robustly optimized BERT pretraining approach. ArXiv:1907.11692

99 Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16×16 words: transformers for image recognition at scale. ArXiv:2010.11929

100 Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision, 2020. 213–229

101 Tanjim M M, Su C, Benjamin E, et al. Attentive sequential models of latent intent for next item recommendation. In: Proceedings of The Web Conference 2020, 2020. 2528–2534

102 Tran Q, Tran L, Hai L C, et al. From implicit to explicit feedback: a deep neural network for modeling sequential behaviours and long-short term preferences of online users. Neurocomputing, 2022, 479: 89–105

103 He T, Li K, Chen S, et al. DMBIN: a dual multi-behavior interest network for click-through rate prediction via contrastive learning. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023. 1366–1375

104 Elsayed S, Rashed A, Schmidt-Thieme L. Multi-behavioral sequential recommendation. In: Proceedings of the 18th ACM Conference on Recommender Systems, 2024. 902–906

105 Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res, 2020, 21: 1–67

106 He M, Pan W, Ming Z. Bar: behavior-aware recommendation for sequential heterogeneous one-class collaborative filtering. Inform Sci, 2022, 608: 881–899

107 Ma M, Na S, Wang H, et al. The graph-based behavior-aware recommendation for interactive news. Appl Intell, 2022, 52: 1913–1929

108 Jia M, Liu F , Li X, et al. Hybrid graph neural network recommendation based on multi-behavior interaction and time sequence awareness. Electronics, 2023, 12: 1223

109 Yu B, Li X, Fang J, et al. Memory-augmented meta-learning framework for session-based target behavior recommendation. World Wide Web, 2023, 26: 233–251

110 Li Q, Ma H, Jin W, et al. Multi-interest network with simple diffusion for multi-behavior sequential recommendation. In: Proceedings of the 2024 SIAM International Conference on Data Mining, 2024. 734–742

111 Xia L, Huang C, Xu Y, et al. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021. 4486–4493

112 Qiao J, Wang L. Modeling user micro-behaviors and original interest via Adaptive Multi-Attention Network for session-based recommendation. Knowl-Based Syst, 2022, 244: 108567

113 Shao Z, Wang S, Lu W, et al. Filter-enhanced hypergraph transformer for multi-behavior sequential recommendation. In: Proceedings of the ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing, 2024. 6575–6579

114 Mukande T, Ali E, Caputo A, et al. A flash attention transformer for multi-behaviour recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023. 4210–4214

115 Wu S, Tang Y, Zhu Y, et al. Session-based recommendation with graph neural networks. In: Proceedings of AAAI Conference on Artificial Intelligence, 2019. 346–353

116 Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. Adv Neural Inform Process Syst, 2020, 33: 6840–6851

117 Wang W, Xu Y, Feng F, et al. Diffusion recommender model. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023. 832–841

118 Li Z, Sun A, Li C. DiffuRec: a diffusion model for sequential recommendation. In: Proceedings of Advances in Neural Information Processing Systems, 2023, 1–28

119 Yang Z, Wu J, Wang Z, et al. Generate what you prefer: reshaping sequential recommendation via guided diffusion. In: Proceedings of Advances in Neural Information Processing Systems, 2024

120 Liu Q, Yan F, Zhao X, et al. Diffusion augmentation for sequential recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023. 1576–1586

121 Wu Z, Wang X, Chen H, et al. Diff4Rec: sequential recommendation with curriculum-scheduled diffusion augmentation. In: Proceedings of the 31st ACM International Conference on Multimedia, 2023. 9329–9335

122 Wang S, Li B Z, Khabsa M, et al. Linformer: self-attention with linear complexity. ArXiv:2006.04768

123 Sun F, Liu J, Wu J, et al. BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019. 1441–1450

124 He X, Deng K, Wang X, et al. LightGCN: simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020. 639–648

125 Dao T, Fu D, Ermon S, et al. Flashattention: fast and memory-efficient exact attention with IO-awareness. Adv Neural Inform Process Syst, 2022, 35: 16344–16359

126 Xie S, Liao Y, Cao K. Multi-behavior recommendation based on heterogeneous graph and attention networks. In: Proceedings of the 4th International Conference on Consumer Electronics and Computer Engineering, 2024. 200–203

127 Qu M, Wang N, Li J. A two-path multibehavior model of user interaction. Electronics, 2023, 12: 3048

128 Li W, Wu Y, Liu Y, et al. BMLP: behavior-aware MLP for heterogeneous sequential recommendation. Front Comput Sci, 2024, 18: 183341

129 Xing X, Zhang X, Cui J, et al. Multi-behavior attention mechanisms graph neural networks based on session recommendation. In: Proceedings of the 35th Chinese Control and Decision Conference, 2023. 4213–4217

130 Shi X, Shi Y, Wang J, et al. Category enhanced dual view contrastive learning for session-based recommendation. In: Proceedings of International Conference on Artificial Neural Networks, 2023. 285–297

131 Qin S, Lin F, Xu L, et al. Multi-behavior session-based recommendation via graph reinforcement learning. In: Proceedings of Asian Conference on Machine Learning, 2024. 1119–1134

132 Elsayed S, Rashed A, Schmidt-Thieme L. HMAR: hierarchical masked attention for multi-behaviour recommendation. In: Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2024. 131–143

133 Gao Y, Huang X, Tao J. Sequential recommendation model integrating micro-behaviors and attribute enhancement. Neurocomputing, 2023, 548: 126393

134 Chen Y, Cao Q, Huang X, et al. Multi-behavior collaborative contrastive learning for sequential recommendation. Complex Intell Syst, 2024, 10: 5033–5048

135 Mukande T, Ali E, Caputo A, et al. A sparse sinkhorn transformer for multi-behaviour recommendation. In: Proceedings of the 2023 IEEE International Conference on e-Business Engineering, 2023. 90–96

136 Xu W, Li H, Wang M. Multi-behavior guided temporal graph attention network for recommendation. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2023. 297–309

137 Zhang C, Chen R, Zhao X, et al. Denoising and prompt-tuning for multi-behavior recommendation. In: Proceedings of the ACM Web Conference 2023, 2023. 1355–1363

138 Wei C, Hu C, Wang C D, et al. Time-aware multibehavior contrastive learning for social recommendation. IEEE Trans Indust Inform, 2024, 20: 6424–6435

139 Liang X, Zhu Y, Duan H, et al. MISR: a multiple behavior interactive enhanced learning model for social-aware recommendation. J Supercomput, 2023, 79: 14221–14244

140 Zhao G, Wang X, Wang J, et al. Task recommendation for mobile crowd sensing system based on multi-view user dynamic behavior prediction. Peer-to-Peer Netw Appl, 2023, 16: 1536–1550

141 Su T T, Xi W D, Xing X X, et al. Mcrec: multi-channel gated gifts recommendation. In: Proceedings of the 2023 IEEE International Conference on Data Mining, 2023. 548–557

142 Gu P, Hu H, Xu G. Modeling multi-behavior sequence via hypergru contrastive network for micro-video recommendation. Knowl-Based Syst, 2024, 298: 111841

143 Pan Y, Gao C, Chang J, et al. Understanding and modeling passive-negative feedback for short-video sequential recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems, 2023. 540–550

144 Liu Q, Zhou Z, Jiang G, et al. Deep task-specific bottom representation network for multi-task recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023. 1637–1646

145 Li J, Sun P, Wang Z, et al. Intent-aware ranking ensemble for personalized recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023. 1004–1013

146 Zhou X, Jin J, Ma L, et al. Script: sequential cross-meta-information recommendation in pretrain and prompt paradigm. In: Proceedings of the 2023 IEEE International Conference on Data Mining, 2023. 888–897

147 Krichene W, Rendle S. On sampled metrics for item recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020. 1748–1757

148 Dallmann A, Zoller D, Hotho A. A case study on sampling strategies for evaluating neural sequential item recommendation models. In: Proceedings of the 15th ACM Conference on Recommender Systems, 2021. 505–514

149 Guo G, Qiu H, Tan Z, et al. Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems. Knowl-Based Syst, 2017, 138: 202–207

150 Li S, Jin X, Xuan Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 5243–5253

151 Hu J, Shen Y, Liu Y, et al. Hero-gang neural model for named entity recognition. ArXiv:2205.07177

152 Lin X, Luo J, Pan J, et al. Multi-sequence attentive user representation learning for side-information integrated sequential recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, 2024. 414–423

153 Huang B, Luo J, Du W, et al. Cascaded cross attention for review-based sequential recommendation. In: Proceedings of the 2023 IEEE International Conference on Data Mining, 2023. 170–179

154 Wang Q, Ji Z, Liu H, et al. Deep Bayesian multi-target learning for recommender systems. ArXiv:1902.09154

155 Ma J, Zhao Z, Yi X, et al. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018. 1930–1939

156 Guo L, Tang L, Chen T, et al. Da-gcn: a domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. In: Proceedings of IJCAI International Joint Conference on Artificial Intelligence, 2021. 2483–2489

157 Li P, Jiang Z, Que M, et al. Dual attentive sequential learning for cross-domain click-through rate prediction. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021. 3172–3180

158 Ma M, Ren P, Chen Z, et al. Mixed information flow for cross-domain sequential recommendations. ACM Trans Knowl Discov Data, 2022, 16: 64

159 Floridi L, Chiriatti M. GPT-3: its nature, scope, limits, and consequences. Minds Mach, 2020, 30: 681–694

160 Zhao W X, Zhou K, Li J, et al. A survey of large language models. ArXiv:2303.18223

161 Wu L, Zheng Z, Qiu Z, et al. A survey on large language models for recommendation. ArXiv:2305.19860

162 Zhao Z, Fan W, Li J, et al. Recommender systems in the era of large language models (LLMs). IEEE Trans Knowl Data Eng, 2024, 36: 6889–6907

163 Wang S, Zhangb X, Wang Y, et al. Trustworthy recommender systems. ArXiv:2208.06265

164 McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of Artificial Intelligence and Statistics, 2017. 1273–1282

165 Yang Q, Liu Y, Chen T, et al. Federated machine learning: concept and applications. ACM Trans Intell Syst Tech, 2019, 10: 1–19