Front. Comput. Sci., 2024, 0(0): 1–45
https://doi.org/10.1007/sxxxxx-yyy-zzzz-1

REVIEW ARTICLE

arXiv:2412.12770v1 [cs.IR] 17 Dec 2024

A Survey on Sequential Recommendation

Liwei Pan[1], **Weike Pan(✉)**[1], **Meiyan Wei**[1], **Hongzhi Yin**[2], **Zhong Ming**[3]

1 College of Computer Science and Software Engineering, Shenzhen University, China
2 School of Electrical Engineering and Computer Science, The University of Queensland, Australia
3 College of Big Data and Internet, Shenzhen Technology University, China

© Higher Education Press 2024

Abstract Different from most conventional recommendation problems, sequential recommendation focuses on learning users' preferences by exploiting the internal order and dependency among the interacted items, which has received significant attention from both researchers and practitioners. In recent years, we have witnessed great progress and achievements in this field, necessitating a new survey. In this survey, we study the SR problem from a new perspective (i.e., the construction of an item's properties), and summarize the most recent techniques used in sequential recommendation such as pure ID-based SR, SR with side information, multi-modal SR, generative SR, LLM-powered SR, ultra-long SR and data-augmented SR. Moreover, we introduce some frontier research topics in sequential recommendation, e.g., open-domain SR, data-centric SR, could-edge collaborative SR, continuous SR, SR for good, and explainable SR. We believe that our survey could be served as a valuable roadmap for readers in this field.

Keywords Sequential Recommendation, ID-based, Side Information, Recent Advancements, New Problems

1 Introduction

Recommender systems are often designed to deliver items that users are interested in from a large collection, which are expected to address the information overload problem and save time for users. Meanwhile, recommender systems are usually deemed effective in improving business profits by motivating people to purchase their items of interest. So far, they have been deployed in various real-world applications, e.g., e-commerce (e.g., Amazon and Alibaba), streaming services (e.g., YouTube and TikTok), social media (e.g., WeChat and Twitter), online advertising, and so on.

Various sequential recommendation (SR) models have been proposed, which have achieved significant improvements in performance recently [1]. The main idea of an SR model is to exploit the position or sequential information of the interacted items, so

Received 12 07, 2024; accepted month dd, yyyy

E-mail: panweike@szu.edu.cn

as to capture the dependencies among them. Given a user's interaction sequence, a typical SR model aims to predict the next likely-to-be-preferred items for this user. In recent years, there has been a significant increase in the number of SR models. By searching the keywords (i.e., sequential, recommend) in the DBLP website[1)], we have the specific publication numbers of papers about sequential recommendation, which are shown in Figure 1. We can see that the number of papers about sequential recommendation increases with time going by. Note that the number of papers published in 2024 is slightly fewer than in 2023, because we collected the data on June 12th, 2024.

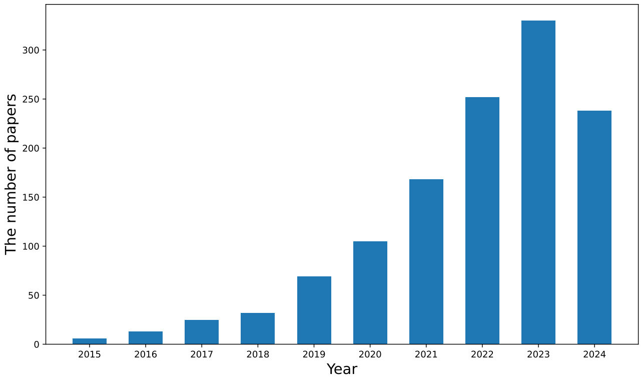

Fig. 1 The number of articles on sequential recommendation published between 2015 and 2024.

Some early SR models [1, 2] use a unique ID to denote an item. These SR models are simple yet effective. For each item, an SR model usually needs to learn a unique item embedding, which is often efficient. Meanwhile, the item embeddings can capture the correlations among the items and extract their latent characteristics. However, these SR models have some weaknesses. For instance, only using item IDs cannot solve cold-start and sparsity problems. Therefore, some SR models [3, 4] combine item IDs and item features. The item features mainly include categorical features (e.g., categories), numerical features (e.g., prices) and graph structure features (e.g., social networks and knowledge graph). Due to privacy issues, we often know little about the users. By leveraging these features, we can effectively learn item representations even when interactions with some items are sparse. Therefore, these SR models can alleviate the cold-start and data sparsity problems. However, these SR models also have some limitations. For example, they usually heavily rely on item IDs during training. If data are different, the item IDs are different. Therefore, the models trained on one dataset cannot be easily transferred to another dataset through fine-tuning or other methods. With the fast development of natural language processing and computer vision, models like BERT [5] and ResNet [6] can extract rich features from text descriptions and images. To solve the issue of data dependency, some SR models [7, 8] leverage BERT and ResNet to extract semantic representations of items from text descriptions and images, respectively, and then directly use them to represent the items. Due to the widespread availability of multi-modal features, these models trained on one data from a certain platform can be deployed on another platform through fine-tuning. However, the semantic representations extracted from multi-modal features cannot fully capture users' fine-grained preferences and collaborative signals. Therefore, some SR models turn to combine the multi-modal features with item IDs to learn users' preferences more comprehensively.

LLMs are trained using a large amount of data and demonstrate promising performance. More importantly, LLMs have a strong reasoning ability. Therefore, LLMs have been applied to many applications, including recommender systems. For example, some SR models leverage LLMs to rec-

[1)] https://dblp.org/

ommend items to users directly [9]. Some SR models leverage the output representations of LLMs for recommending personalized items more accurately since the embeddings contain rich semantic knowledge [10]. To alleviate data sparsity and cold-start problems, some SR models leverage LLMs to generate some data. These SR models then combine the generated data with the original data to achieve better recommendation performance. However, LLMs also have some shortcomings. For example, training LLMs-based SR models requires significant computational resources. This survey summarizes how to leverage LLMs in SR models.

In sequential recommendation, modeling ultra-long interaction sequences becomes increasingly complex. Over time, users interact with more and more items. With the length of interaction sequences increasing, the time of both training and inference will increase. Meanwhile, as users interact with an increasing number of items, noise also becomes more serious in the interaction sequences. Therefore, modeling ultra-long interaction sequences will become increasingly difficult. Common approaches retrieve a few interacted items from ultra-long interaction sequences [11]. Our surveys will summarize some common methods applied to ultra-long interaction sequences modeling.

In our survey, we also introduce data augmentation methods in SR models. This is because data augmentation methods can be leveraged by different SR models. They mainly focus on generating more interaction data to solve data sparsity and cold-start problems. Common augmentation methods include some operations, such as resorting, masking, cropping, and others on interaction sequences [12].

The contributions of our survey are summarized as follows:

- Our survey summarizes the early and recent works about SR models from a more comprehensive manner.
- Our survey categorizes the existing SR models into four categories according to the construction of item properties.
- Our survey summarizes the latest techniques applied to sequential recommendation
- Our survey introduces empirical studies and future research directions in SR models.

2 Related Surveys

In recent years, an increasing number of surveys on sequential and session-based recommender systems have been published [13–18]. They introduce some common algorithms and models for these systems. However, with the development of LLMs and other techniques, these surveys are out-of-date to some extent as LLMs have strong deduction abilities and can fully extract semantic information for recommendation. Meanwhile, some SR models introduced in these surveys still have weaknesses. For example, only using an item ID to denote an item might influence the transferability of SR models. Some surveys [19, 20] focus on cross-domain sequential recommender systems, or multi-behavior sequential recommender systems. They are only a small fraction of models in SR models. Compared with these surveys, our survey introduces the development of sequential recommender systems more comprehensively. There are also some surveys about recommender systems. Jing et al. [21] introduce contrastive learning in recommender systems comprehensively. Lai et al. [22] discuss the importance of data and some methods of data augmentation. Zhang et al. [23] explore the usage of different kinds of side information in recommender systems. However, these surveys only focus on one specific certain technique in recommender systems.

Compared with these surveys, we not only classify existing research on SR but also introduce a new taxonomy to comprehend, structure and analyze these works.

3 Sequential Recommendation

In this section, we will begin by defining the problem of sequential recommendation. Then, we will introduce properties about users, items, behaviors and sequences as well as different kinds of features. Next, we will describe four different types of SR models. Finally, we will discuss some challenges faced by SR models.

3.1 Problem Definition

In sequential recommendation, we often have one or more sequences of interacted items w.r.t. each user, as well as some auxiliary information to help learn user preferences. Our goal is then to generate a ranked list of items accurately for each user.

3.2 Properties

In sequential recommendation systems, there are two important entities (i.e., an item and a user). Therefore, item properties and user properties are vital for SR models. A user interacts with items, and the user and the items are connected according to behaviors. The user's interaction sequence is constructed according to interaction order. Therefore, the properties of both behaviors and sequences are also important in sequential recommendation systems. The specific details about these properties in sequential recommendation systems are shown in Figure 2.

For item properties in SR, each item is denoted by a unique number (i.e., an item ID). Apart from the item ID, each item also has side information. The side information mainly has two types, including multi-modal features and general features. The general features have three types: categorical features, numerical features, and knowledge graph. The categorical features can be denoted by one-hot vectors. The numerical features are values represented by numbers, which have real meaning. Each numerical feature usually lies in a continuous or discrete range. User properties consist of two parts: a unique user ID and some side information specific to each user. Some specific details are shown in Figure 2. To avoid the leakage of users' privacy, the side information about users is usually unseen. Behavior properties are also comprised of general features and multi-modal features. Some features such as ratings and reviews, which might not be present in certain interactions, because not every user can provide a review or rating for an item they interact with. However, if a user interacts with an item, the behavior type of this interaction is bound to exist. Context information refers to an environment in which a behavior happens, such as a specific time and location. Sequential properties primarily include the length of interaction sequences and the internal order of items within these sequences. Meanwhile, there exist two important graph-structure features (i.e., knowledge graph and social networks).

Entity IDs (i.e., item IDs and user IDs), general features and multi-modal features are quietly different in sequential recommender systems. The entity IDs are specific to a particular dataset. In different datasets or different platforms, the encoding methods for the entity IDs are quite different. For example, a movie might be encoded as item 1 in the MovieLens dataset [2)]. However, a book might also be encoded as item 1 in the book domain of

[2)] https://grouplens.org/datasets/movielens/

Table 1 Some notations used in the paper.

Notation	Description
$\mathcal{U} = \{u_1, u_2, \dots, u_{\lvert U \rvert}\}$	A set of users
$\mathcal{I} = \{i_1, i_2, \dots, i_{\lvert I \rvert}\}$	A set of items
L	The length of an interaction sequence
$S_u = \{s_{u,1}, s_{u,2}, \dots, s_{u,L}\}$	The interaction sequence of user u
$F_u^{(k)} = \{f_{u,1}^{(k)}, f_{u,2}^{(k)}, \dots, f_{u,L}^{(k)}\}$	The k-th multi-modal feature w.r.t. user u's interaction sequence.
$N_u^{(k)} = \{n_{u,1}^{(k)}, n_{u,2}^{(k)}, \dots, n_{u,L}^{(k)}\}$	The k-th categorical and numerical feature w.r.t. user u's interaction sequence.
K_m	The number of multi-modal features
K_g	The number of categorical and numerical features
t_u	The item that user u interacts with at the most recent time step
$R_{u,i}$	The rank generated by an SR model of candidate item i for user u
$P(i \lvert S_u)$	The probability in which user u interacted item i given historical sequence S_u
GNNs(·), CNNs(·), Transformer(·)	The GNNs, CNNs and Transformer models
Embed(·)	The item embedding layer
$\mathbf{P}_u^{CNNs}$, $\mathbf{P}_u^{GNNs}$	The user u's preferences extracted from CNNs and GNNs
$\mathbf{P}_u^{Tr}$	The user u's preferences extracted from Transformer
$\mathbf{P}_u^{final}$	The user u's final preferences
$\mathcal{G}$	The graph constructed by all users' interaction sequences
Fuse(·)	The fusion module
$\mathrm{BM}_k(\cdot)$	The k-th backbone model
$\mathbf{E}_u$	The item embedding matrix for user u
$\mathbf{A}_u^{(k)}$	The k-th feature embedding matrix for user u
$S_{u,1}^{(k)}, S_{u,2}^{(k)}, \dots, S_{u,m}^{(k)}$	The m sub-sequences divided by the k-th feature
$\mathbf{P}_u^{feat}$	the u's preferences extracted from the features of interacted items
Text$_i$, Img$_i$, Vid$_i$	The text descriptions, images and videos of item i
TextEncoder(), ImageEncoder(), VideoEncoder(·)	The text encoder, image encoder and video encoder
$\mathbf{e}_i$	The representations about item i
$\mathbf{Y}_i = \{y_{i,1}, y_{i,2}, \dots, y_{i,l}\}$	The item i denoted by l tokens
x_{prompt}	The prompt which is fed into LLMs
LLMs(·)	The large language models
$\hat{x}_{prompt}$	The outputs of LLMs while inputting x_{prompt}
$\mathbf{P}_u^{short}$,$\mathbf{P}_u^{long}$	The user u's short-term and long-term interests
Att(·),Ret(·)	The attention and retrieval modules
$S_{u,recent}$	The user u's recently behavior sequence

Amazon dataset [3]. Therefore, item embeddings based on item IDs cannot be transferred from one platform to another platform. The general features share some common characteristics across different datasets or platforms. For example, basic behavior types in different datasets might be the same. All e-commerce platforms contain basic behavior types (i.e., click and purchase). However, the categories of items might differ across different domains or platforms, leading to different distributions of general features. If we apply the embeddings of general features in one domain to another domain, it might result in negative transfer. Techniques like a cross-attention mechanism can address this issue to some

[3] https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023

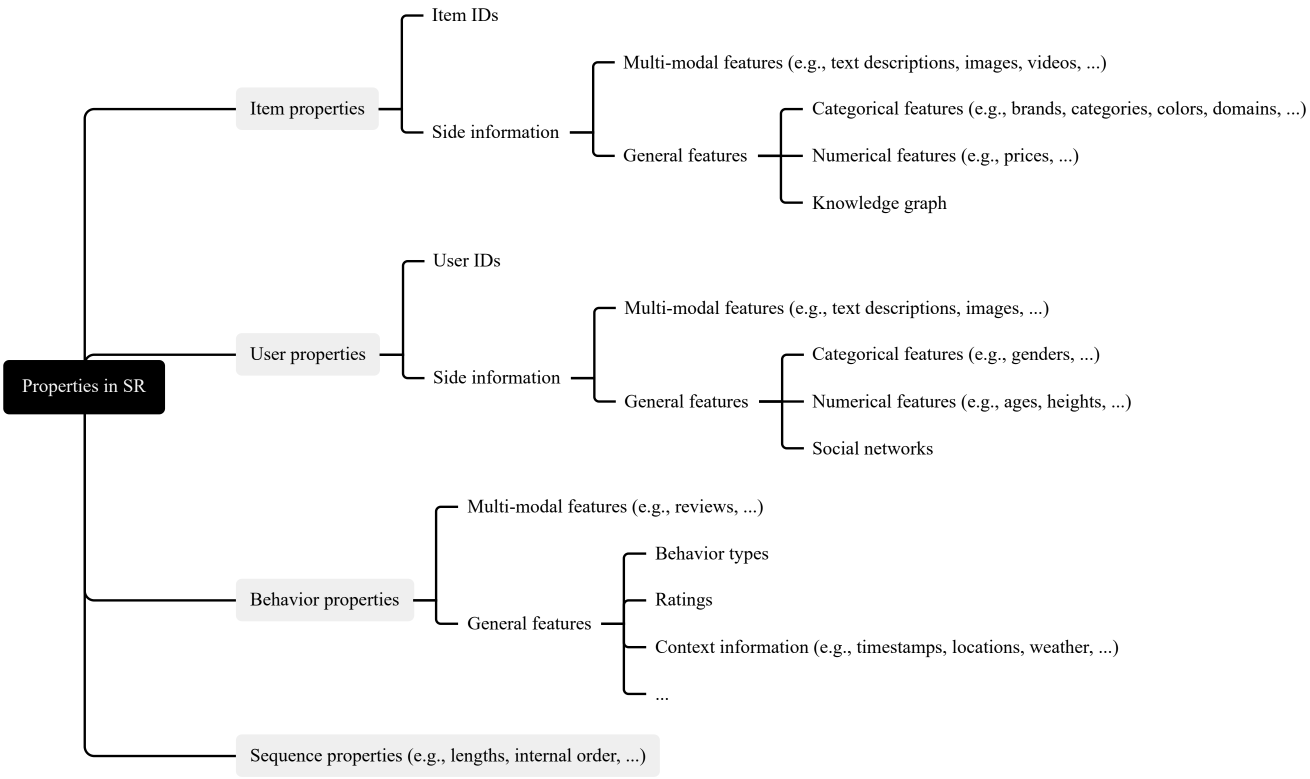
Properties in SR
Item properties
Item IDs
Side information
Multi-modal features (e.g., text descriptions, images, videos, ...)
General features
Categorical features (e.g., brands, categories, colors, domains, ...)
Numerical features (e.g., prices, ...)
Knowledge graph
User properties
User IDs
Side information
Multi-modal features (e.g., text descriptions, images, ...)
General features
Categorical features (e.g., genders, ...)
Numerical features (e.g., ages, heights, ...)
Social networks
Behavior properties
Multi-modal features (e.g., reviews, ...)
General features
Behavior types
Ratings
Context information (e.g., timestamps, locations, weather, ...)
...
Sequence properties (e.g., lengths, internal order, ...)

Fig. 2 Properties in sequential recommendation.

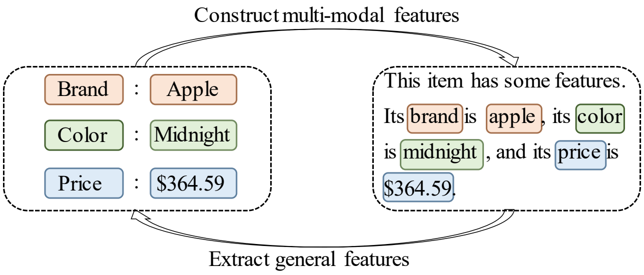
Construct multi-modal features
Brand : Apple
Color : Midnight
Price : $364.59
This item has some features.
Its brand is apple, its color is midnight, and its price is $364.59.
Extract general features

Fig. 3 The conversion between multi-modal features and general features.

extent. Multi-modal features, such as text, images, and videos, are universal across all domains and platforms. We use language to record information as text, cameras to capture beautiful images, videos to document moments. The text, images, and videos have separate embedding spaces. Meanwhile, they all have shared space. The multi-features can be leveraged in all domains and all platforms. If we leverage the text descriptions of an item to represent the item in a model and train the model on one platform, it can be directly applied to another platform or fine-tuned as needed. Therefore, leveraging multi-modal features to denote an item can fully realize the knowledge transfer from one platform to another.

In the meantime, the general features and multi-modal features can be converted into each other. As shown in Figure 3, we can convert an item's brand, price and color into text descriptions. Then, SR models can leverage these text descriptions of this item to learn the universal representations of this item. However, this type of conversion has shortcomings. For example, if we convert timestamps to text, fine-grained time information might be damaged. Meanwhile, as shown in Figure 3, we can also extract general features from the text descriptions of

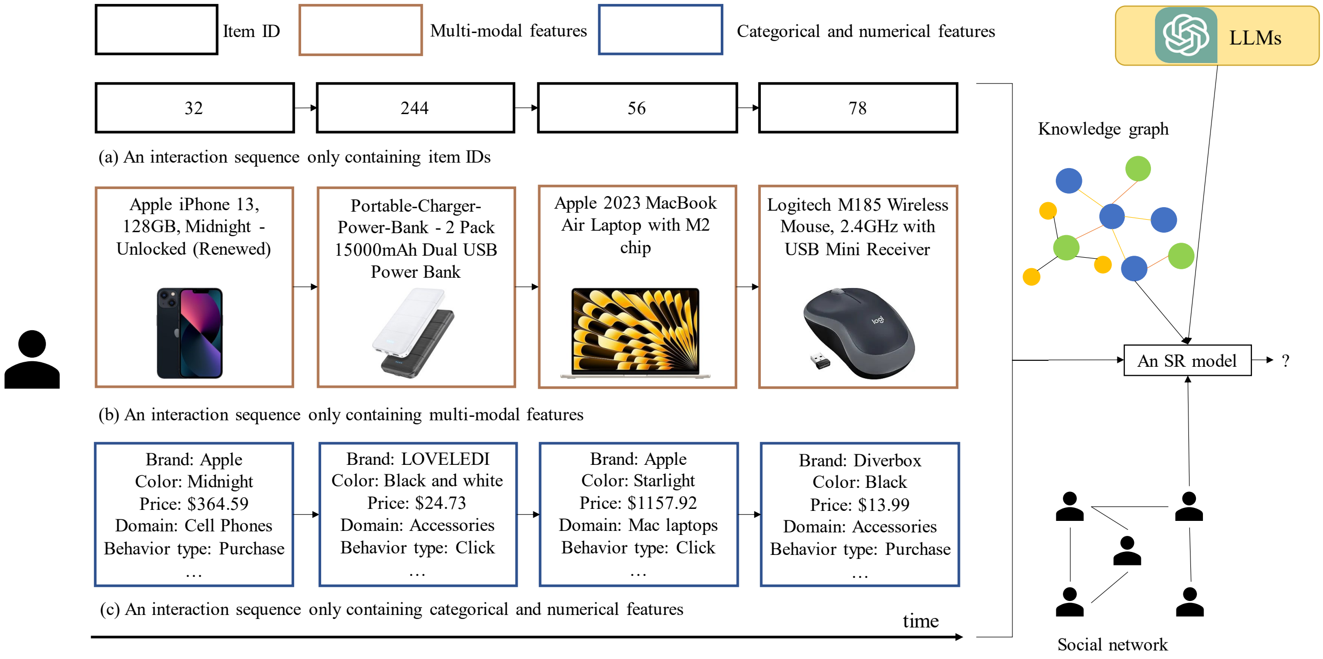
Item ID
Multi-modal features
Categorical and numerical features
LLMs
32
244
56
78
(a) An interaction sequence only containing item IDs
Knowledge graph
Apple iPhone 13, 128GB, Midnight - Unlocked (Renewed)
Portable-Charger-Power-Bank - 2 Pack 15000mAh Dual USB Power Bank
Apple 2023 MacBook Air Laptop with M2 chip
Logitech M185 Wireless Mouse, 2.4GHz with USB Mini Receiver
An SR model
?
(b) An interaction sequence only containing multi-modal features
Brand: Apple
Color: Midnight
Price: $364.59
Domain: Cell Phones
Behavior type: Purchase
...
Brand: LOVELEDI
Color: Black and white
Price: $24.73
Domain: Accessories
Behavior type: Click
...
Brand: Apple
Color: Starlight
Price: $1157.92
Domain: Mac laptops
Behavior type: Click
...
Brand: Diverbox
Color: Black
Price: $13.99
Domain: Accessories
Behavior type: Purchase
...
(c) An interaction sequence only containing categorical and numerical features
time
Social network

Fig. 4 An illustration of sequential recommendation.

an item, such as its brand, color and price. However, the extracted features might contain some noise.

There are several important features in sequential recommender systems. Different features have various meanings and play different roles in sequential recommendation. Some specific details are shown as follows:

- Timestamp refers to the precise time when a user interacts with an item. Leveraging timestamp information can capture the user's evolving interests and periodic behaviors precisely.
- Category refers to the hierarchical classification of an item. Leveraging the category of the item can capture users' preferences more precisely and narrow the range of candidate items.
- Price refers to the amount of money spent to purchase an item. It plays a dominant role in deciding whether to purchase the item.
- Text description refers to the description of an item and the title of the item. It can describe the item from different perspectives.
- Image refers to the visual representations of an item. It can provide a clear and intuitive understanding of the item. By looking at images, we can know what the item looks like.
- Rating refers to the degree of a user's satisfaction with an item. It can reflect the user's preferences for an item and the quality of the item.
- Review refers to comments that a user provides for an item. It can offer more detailed information about a user's preferences for the item, and it can reflect the overall quality of this item.
- Behavior type refers to a specific behavior type in which a user interacts with an item, such as click, favorite and purchase. Leveraging different types of behaviors can capture the user's different preferences from different perspectives.

3.3 Categories

In our surveys, we classify SR models into four categories based on the properties of items. As depicted in Figure 4, for user u, there are three different interaction sequences containing item IDs and different features, respectively.

As shown in part (a) of Figure 4, some early works leverage a unique item ID to denote an item for recommendation [1,24]. These SR models leverage only item ID interaction sequences, which can be formalized as: $P(i|S_u)$. In these works, given a candidate item i and a user u's historical interaction sequence S_u, these models need to calculate the probability $P(i|S_u)$ in which user u interacts with the candidate item i.

However, SR models based on pure item IDs cannot tackle cold-start problems well. General features are important for recommending personalized items, which can alleviate the cold-start problems and fully extract users' preferences. As depicted in part (a) and part (c) of Figure 4, some works leverage both item ID sequences and interaction sequences containing general features (i.e., categorical features and numerical features) to recommend personalized items to users [4,25]. These SR models can be formalized as: $P(i|S_u, N_u^{(1)}, N_u^{(2)}, \ldots, N_u^{(K_g)})$. Given a user u's interaction sequence S_u, which contains item IDs and general features $N_u^{(1)}, \ldots, N_u^{(K_g)}$ embedded in the sequence, these models aim to predict the probability that the user u interacted with a candidate item i. To account for the influence of users' friends, social networks might be leveraged in these works [26]. The general features of interacted items can be represented as knowledge graph. Therefore, these works might leverage knowledge graph to capture relationships among general features.

Though the above-mentioned models achieve significant performance, they still have certain shortcomings. Firstly, these models heavily rely on item IDs, which hinders their transferability. Secondly, these general features cannot utilize some of latest techniques effectively, such as LLMs. Therefore, some recently proposed models leverage the multi-modal features of an item to denote the item directly [8,27]. These models can realize the transferability across different platforms. By leveraging latest techniques like LLMs, they can fully extract some semantic knowledge. These models can be formalized as: $P(i|F_u^{(1)}, F_u^{(2)}, \ldots, F_u^{(K_m)})$. Given the multi-modal features derived from the items within a user u's interaction sequence, these models can calculate the probability in which user u interacts with a candidate item i. The core ideas of these models are illustrated in part (b) of Figure 4. When both sequence information and multi-modal features (i.e., text descriptions and images) of each interaction are available, an SR model predicts the next interacted item interacted with by user u. By replacing item IDs with text descriptions or images, SR models can require rich semantic knowledge and enhance their transferability.

However, relying solely on items' multi-modal features cannot capture users' fine-grained interests. In a warm-start situation, only leveraging item IDs can achieve better performance than only leveraging multi-modal features [7]. Therefore, some SR models leverage both item IDs and multi-modal features of these items. The simple diagram of these models is shown by combining part (a) and part (b) in Figure 4. In these SR models, each item property contains both a unique item ID and multi-modal features. To extract semantic information from multi-modal features, LLMs can be leveraged. These SR models can be formalized as:

$P(i|S_u, F_u^{(1)}, F_u^{(2)}, \ldots, F_u^{(K_m)})$.

3.4 Taxonomy

Our survey introduces SR models through a comprehensive taxonomy. As shown in Figure 5, our survey will introduce SR models from three different aspects (i.e., pure ID-based SR, SR with side information and recent SR advancements). Our survey will discuss specific details about these SR models in following sections.

3.5 Challenges

In sequential recommender systems, there are some challenges as follows:

- Cold-start and data sparsity problems: A new user and a new item usually have sparse interactions. It is a challenge to improve recommendation performance in a new user and a new item.
- Ultra-long interaction sequences modeling: Over time, some users might generate ultra-long interaction sequences. It is a challenge to model these ultra-long interaction sequences effectively and efficiently.
- Dynamic interests: Users' interests evolve over time. It is a challenge to capture users' current and long-term interests.
- Diverse interests: Users exhibit diverse interests across different items. It is a challenge to capture users' diverse interests.
- Knowledge transfer across platforms: Items might be quiet different across different platforms. It is a challenge to achieve knowledge transfer across different platforms.

4 Pure ID-based SR

In SR models based on pure IDs, each item and user are represented by a unique item identity (i.e., item ID) and a unique user identity (i.e., user ID), respectively. In the past few years, the number of SR models based on pure IDs has increased significantly. We categorize them into different classes according to the used techniques. In this section, we will first introduce some SR methods using traditional machine learning. Then, we will introduce some SR methods integrating various deep learning techniques. Next, we will introduce some SR methods using reinforcement learning. Finally, we will introduce some SR methods that combine different kinds of deep learning techniques. In this section, all SR models are based on pure IDs.

4.1 Traditional Models

Traditional SR models do not use deep learning techniques. Though these models are simple and effective, they cannot capture long-term sequential dependencies. The traditional SR models can mainly be classified into three classes: frequent pattern mining, Markov models and latent factor models.

FPM (frequent pattern mining) extracts different kinds of patterns from interaction sequences [28]. These models are of good explainability, but they cannot tackle complex datasets. If the dataset is too large, the number of extracted sequential patterns might become excessively too large.

Markov models have an assumption that a predicted action relies on a few recent actions. For example, FPMC [29] predicts the next interacted item for each user by considering both the last interacted item and personalized interests. These models cannot exploit long-term sequential dependencies. If order dependencies are weak in a dataset, Markov models might perform poorly.

Latent factor models mainly learn latent representations to estimate unobserved transitions. TransRec [2] proposes a transition operator: a previously interacted item + a user ≈ the user next interacted item. The transition operator might vary across different datasets. Meanwhile, these SR models cannot capture complex transition relations.

4.2 Deep Learning Models

4.2.1 Recurrent Neural Networks

Compared with traditional SR models, RNNs-based SR models not only can capture short-term sequential dependencies, but also can capture long-term sequential dependencies and non-linear dynamics. In recent years, some RNNs-based SR models have been proposed. GRU4Rec [24] applies RNNs to capture long-term dependencies in session-based data. Based on GRU4Rec, GRU4Rec+ [30] proposes a novel negative item sampling and a novel ranking loss function. Some RNNs-based SR models are designed to capture each sequence independently, thus ignoring the global structure of all sequences. To solve this problem, KrNN-P [31] incorporates neighbor sequences into RNNs to capture global relationships while refining local ones. To consider the dependencies among users' sequences, MrRNN [32] introduces manifold regularization into RNNs based on the multi-facets of collaborative filtering. However, these RNN-based models have some shortcomings. For example, they cannot avoid vanishing gradients while handling ultra-long interaction sequences.

4.2.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) have been widely used in image processing. They mainly consist of two operations (i.e., a convolution operation and a pooling operation). Compared with multi-layer perceptron (MLPs), CNNs can extract features from images more effectively. By regarding user u's interaction sequence as an image, CNNs can also be applied to sequential recommendation. CNNs usually have multiple convolutional filters. Therefore, CNNs-based SR models can extract multi-faceted interests from user u's interaction sequence.

In recent years, several CNNs-based SR models have been proposed. DACNN [33] leverages CNNs to capture local sequential features while considering item sequences. Caser [34] leverages convolutional filters to capture both general preferences and sequential patterns. CosRec [35] leverages 2D convolutional neural networks to model complex interaction relationships. However, CNN-based SR models have certain shortcomings. For example, CNN-based SR models might have a large number of convolutional filters, which can result in excessive computational resource consumption.

4.2.3 Transformer

Transformer has accomplished great achievements in natural language processing, largely due to its self-attention mechanism. The self-attention mechanism can identify crucial information by calculating attention scores.

Transformer architecture has two types: single-direction Transformer (i.e., uni-Transformer) and double-direction Transformer (i.e., bi-Transformer). SASRec [1] is currently the most popular SR model, which leverages the uni-Transformer. SASRec can predict the next interacted items based on historical interacted items. BERT4Rec [36] leverages bi-Transformer, which can predict the next interacted items according to historical interacted items and future interacted items. ALBERT4Rec and De-

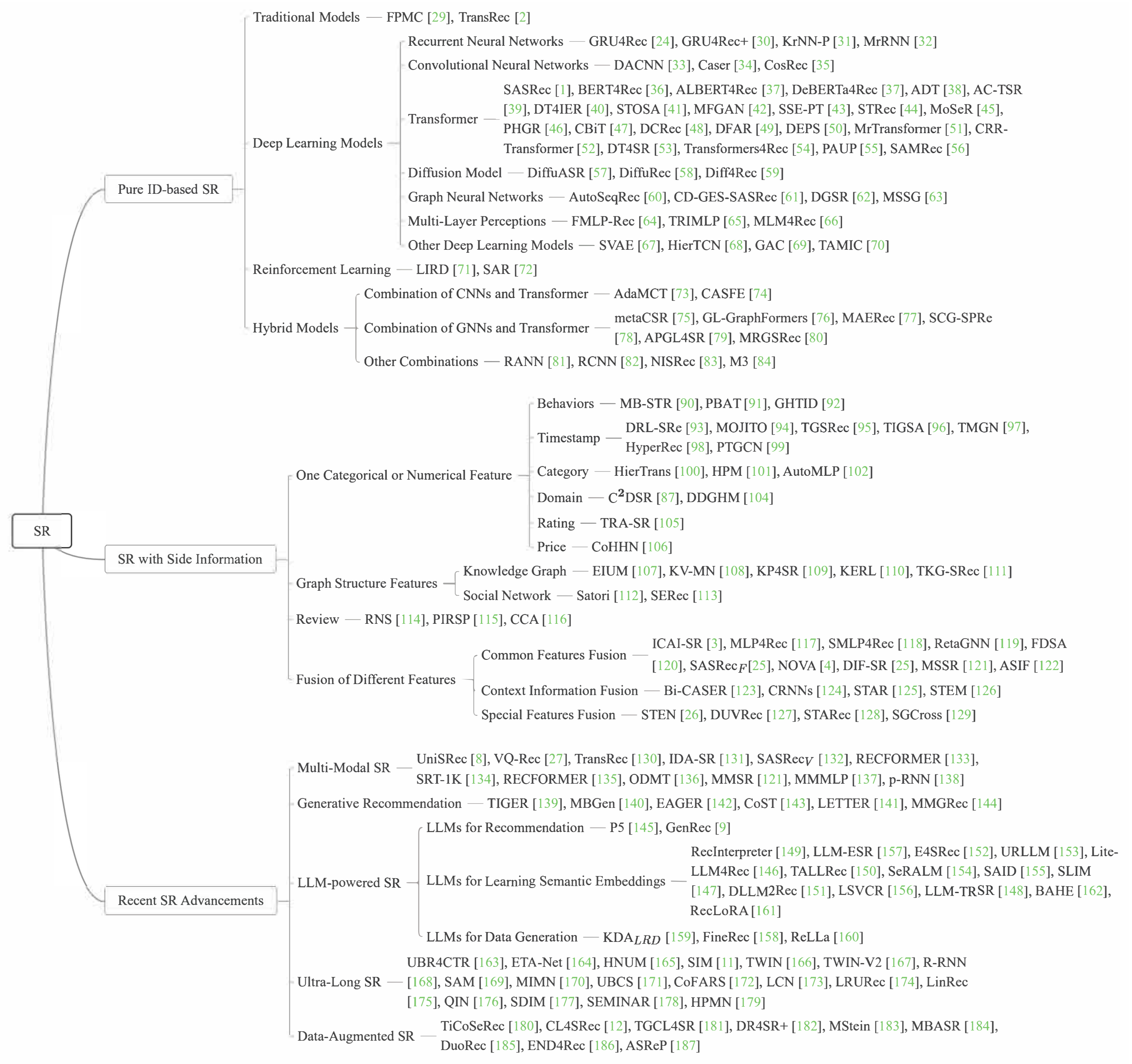
SR
Pure ID-based SR
Traditional Models — FPMC [29], TransRec [2]
Deep Learning Models
Recurrent Neural Networks — GRU4Rec [24], GRU4Rec+ [30], KrNN-P [31], MrRNN [32]
Convolutional Neural Networks — DACNN [33], Caser [34], CosRec [35]
Transformer — SASRec [1], BERT4Rec [36], ALBERT4Rec [37], DeBERTa4Rec [37], ADT [38], AC-TSR [39], DT4IER [40], STOSA [41], MFGAN [42], SSE-PT [43], STRec [44], MoSeR [45], PHGR [46], CBiT [47], DCRec [48], DFAR [49], DEPS [50], MrTransformer [51], CRR-Transformer [52], DT4SR [53], Transformers4Rec [54], PAUP [55], SAMRec [56]
Diffusion Model — DiffuASR [57], DiffuRec [58], Diff4Rec [59]
Graph Neural Networks — AutoSeqRec [60], CD-GES-SASRec [61], DGSR [62], MSSG [63]
Multi-Layer Perceptions — FMLP-Rec [64], TRIMLP [65], MLM4Rec [66]
Other Deep Learning Models — SVAE [67], HierTCN [68], GAC [69], TAMIC [70]
Reinforcement Learning — LIRD [71], SAR [72]
Hybrid Models
Combination of CNNs and Transformer — AdaMCT [73], CASFE [74]
Combination of GNNs and Transformer — metaCSR [75], GL-GraphFormers [76], MAERec [77], SCG-SPRe [78], APGL4SR [79], MRGSRec [80]
Other Combinations — RANN [81], RCNN [82], NISRec [83], M3 [84]
SR with Side Information
One Categorical or Numerical Feature
Behaviors — MB-STR [90], PBAT [91], GHTID [92]
Timestamp — DRL-SRe [93], MOJITO [94], TGSRec [95], TIGSA [96], TMGN [97], HyperRec [98], PTGCN [99]
Category — HierTrans [100], HPM [101], AutoMLP [102]
Domain — C^{2}DSR [87], DDGHM [104]
Rating — TRA-SR [105]
Price — CoHHN [106]
Graph Structure Features
Knowledge Graph — EIUM [107], KV-MN [108], KP4SR [109], KERL [110], TKG-SRec [111]
Social Network — Satori [112], SERec [113]
Review — RNS [114], PIRSP [115], CCA [116]
Fusion of Different Features
Common Features Fusion — ICAI-SR [3], MLP4Rec [117], SMLP4Rec [118], RetaGNN [119], FDSA [120], SASRec$_F$[25], NOVA [4], DIF-SR [25], MSSR [121], ASIF [122]
Context Information Fusion — Bi-CASER [123], CRNNs [124], STAR [125], STEM [126]
Special Features Fusion — STEN [26], DUVRec [127], STARec [128], SGCross [129]
Recent SR Advancements
Multi-Modal SR — UniSRec [8], VQ-Rec [27], TransRec [130], IDA-SR [131], SASRec$_V$ [132], RECFORMER [133], SRT-1K [134], RECFORMER [135], ODMT [136], MMSR [121], MMMLP [137], p-RNN [138]
Generative Recommendation — TIGER [139], MBGen [140], EAGER [142], CoST [143], LETTER [141], MMGRec [144]
LLM-powered SR
LLMs for Recommendation — P5 [145], GenRec [9]
LLMs for Learning Semantic Embeddings — RecInterpreter [149], LLM-ESR [157], E4SRec [152], URLLM [153], Lite-LLM4Rec [146], TALLRec [150], SeRALM [154], SAID [155], SLIM [147], DLLM2Rec [151], LSVCR [156], LLM-TRSR [148], BAHE [162], RecLoRA [161]
LLMs for Data Generation — KDA$_{LRD}$ [159], FineRec [158], ReLLa [160]
Ultra-Long SR — UBR4CTR [163], ETA-Net [164], HNUM [165], SIM [11], TWIN [166], TWIN-V2 [167], R-RNN [168], SAM [169], MIMN [170], UBCS [171], CoFARS [172], LCN [173], LRURec [174], LinRec [175], QIN [176], SDIM [177], SEMINAR [178], HPMN [179]
Data-Augmented SR — TiCoSeRec [180], CL4SRec [12], TGCL4SR [181], DR4SR+ [182], MStein [183], MBASR [184], DuoRec [185], END4Rec [186], ASReP [187]

Fig. 5 Taxonomy of existing works on sequential recommendation.

BERTa4Rec [37] demonstrate superior performance and training efficiency compared to BERT4Rec [36]. ADT [38] learns disentangled diverse interests using a Transformer-based encoder and decoder. However, the large weights of attention mechanisms might not be accurate. Additionally, position encoding and noise inputs might negatively impact the recommendation performance. Therefore, AC-TSR [39] leverages contrastive learning to achieve noise reduction. DT4IER [40] leverages decision Transformer to solve user retention challenges. Due to the uncertainty of users' sequential behaviors, STOSA [41] leverages stochastic Gaussian distribution to embed items and introduces a novel Wasserstein self-attention module to capture item-item relationships. MFGAN [42] uses a Transformer-based generator to recommend the next possible items and multiple discriminators to evaluate generated sub-sequences. SSE-PT [43] introduces user embeddings to realize the personalization of Transformer. STRec [44] leverages sparse Transformer to focus on the most relevant interactions while predicting target items. Meanwhile, STRec [44] replaces the self-attention mechanism with a cross-attention mechanism. MoSeR [45] captures motifs hidden in behavior sequences to model micro-structure features. PHGR [46] leverages a novel hyperbolic inner product operator to enable global (all user-item interactions) and local (each user's interactions) graph representation learning in Poincaré ball. CBiT [47] leverages contrastive learning and bi-Transformer to learn more fine-grained preferences. DCRec [48] leverages cross-view contrastive learning to capture correlations among different users' interaction sequences. DFAR [49] leverages a dual-interest disentangling layer to disentangle positive and negative interests for learning users' transition patterns better. DEPS [50] leverages both users' and items' interaction sequences as the inputs to two Transformers for learning preferences better. MrTransformer [51] leverages a preference separation module to learn both users' common and unique preferences. CRR-Transformer [52] applies a pre-trained transformer model to an online RL algorithm. Due to the uncertainty of interactions, DT4SR [53] leverages an elliptical Gaussian distribution to describe items and adopts Wasserstein distance to measure the similarity among distributions. Transformers4Rec [54] applies Transformer-based architectures for recommendation, such as GPT-2, XLNET, and Transformer-XL. PAUP [55] captures both short- and long-term patterns through a progressive attention distribution mechanism. In this model, long behavior sequences are segmented into a series of sub-sequences using a down-sampling convolution module. SAMRec [56] enhances the recommendation performance of Transformer from the perspective of loss geometry.

Though Transformer-based models have achieved significant progress, these models still have some shortcomings. For example, Transformer-based SR models suffer from a quadratic computational complexity.

4.2.4 Diffusion Model

Diffusion model has been extensively applied to various generative tasks, such as image generation.

In recent years, several SR models based on diffusion model have been proposed. DiffuASR [57] leverages a diffusion-based framework to generate high-quality interactions guaranteeing the similarity between generated and original sequences. DiffuRec [58] is the first work that applies the diffusion model to sequential recommendation. By corrupting and reconstructing target item embeddings, DiffuRec [58] injects uncertainty during the recommendation process. Diff4Rec [59] leverages the diffusion model to augment data by adopting a curriculum scheduling strategy, including interaction augmentation and objective augmentation. Diffusion-based SR models have some weaknesses. Compared with Transformer-based SR models, diffusion-based SR models require more computational resources and training time.

4.2.5 Graph Neural Networks

GNNs-based SR models can learn complex high-order interactions and capture collaborative information. Therefore, GNNs-based SR models have received widespread attention. AutoSeqRec [60] leverages an encoder and three decoders to reconstruct user-item interaction and transition matrices. CD-GES-SASRec [61] leverages a causal graph to distinguish causal and non-causal transitions. DGSR

[62] constructs a user-item interaction graph to capture personal preferences and an item-item transition graph to model transitional relationships between adjacent items. Meanwhile, DGSR [62] injects high-order connections into graphs. MSSG [63] transforms items within a sequence into a star graph. Meanwhile, MSSG introduces an additional internal node to capture global information. MSSG overcomes the over-smoothing issue and realizes a linear time complexity. However, GNNs-based SR models might not fully capture sequential patterns and can be computationally expensive to train.

4.2.6 Multi-Layer Perceptions

Compared with Transformer, leveraging MLPs to calculate is more efficient (i.e., a linear time complexity). In recent years, some MLPs-based SR models have been proposed. FMLP-Rec [64] incorporates learnable filters into MLPs architecture for reducing noise. TRIMLP [65] disables triangle neurons in MLPs to account for chronological order in users' interaction sequences. MLM4Rec [66] learns users' global and local preferences by incorporating convolution operators into MLPs. However, the MLPs-based SR models struggle to capture complex sequential patterns.

4.2.7 Other Deep Learning Models

Variational autoencoder (VAE) can learn compressed features from input embeddings. SVAE [67] combines both VAE and RNNs to capture sequential dependencies, with VAE extracting features at each time step.

Temporal convolutional network (TCN) is a specialized type of convolutional neural network. Compared with RNNs, TCN is more computationally efficient. HierTCN [68] combines both TCNs and RNNs to learn short-term interests and long-term interests in sequences.

Capsule networks can recognize objects from different perspectives. GAC [69] leverages capsule networks to model personalized item-level and factor-level sequential dependencies, and combines the two kinds of sequential dependencies to recommend items to users. TAMIC [70] utilizes a time-aware dynamic routing algorithm and two kinds of time-aware voting gates to inject temporal information into sequential modeling.

4.3 Reinforcement Learning

Reinforcement learning can efficiently model interactions between users and a recommender system. LIRD [71] regards sequential interactions between the users and the recommender system as a Markov decision process (MDP), and leverages reinforcement learning to learn optimal strategies for recommending items automatically. SAR [72] leverages reinforcement learning to learn dependencies and the length of the interaction sequence for each user.

4.4 Hybrid Models

Some SR models demonstrate significant improvements in recommendation performance by combining different techniques. For example, combining Transformer and GNNs can achieve better performance. As shown in Table 2, each technique offers distinct advantages. Therefore, the combined models can combine the strengths of different techniques. The combined models can be roughly categorized into three classes (i.e., CNNs + Transformer, GNNs + Transformer and other combined types). Integrating various neural network architectures enables SR models to achieve enhanced recommendation performance. Several commonly used

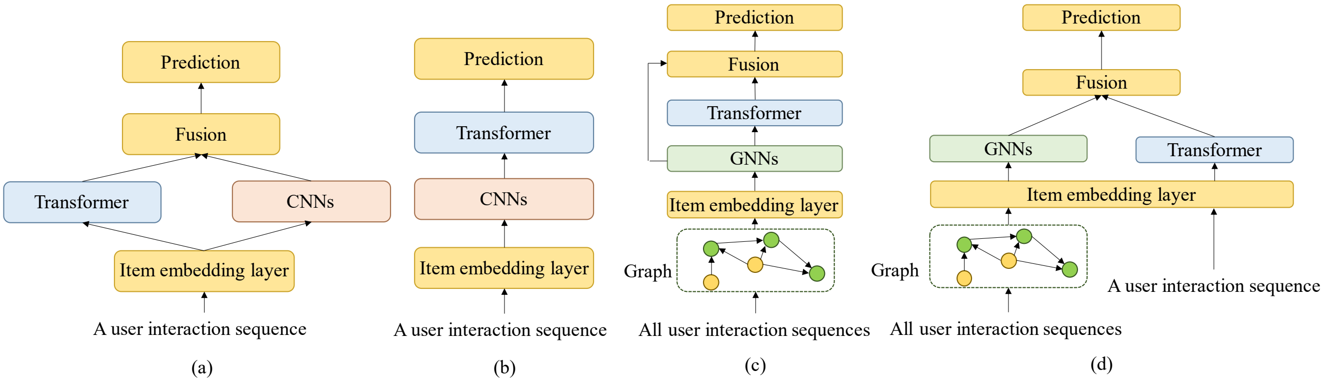

Fig. 6 Illustration of the common combination strategies in sequential recommendation.

combined models are depicted in Figure 6. As depicted in Figure 6, there are four kinds of combined types. Two different techniques can be leveraged in parallel. For example, CNNs and Transformer can extract short-term dependencies and long-term dependencies from user interaction sequences, respectively. And then the two kinds of dependencies are fused together. Alternatively, the two techniques can be used in a cascaded manner. For example, GNNs can be denoted as a feature extractor for extracting features, and the extracted features can then be fed into Transformer.

4.4.1 Combination of CNNs and Transformer

Transformer and CNNs exhibit unique abilities to capture different types of dependencies. Transformer has a good ability to capture long-term dependencies. CNNs are effective at modeling short-term dependencies and multiple-faced preferences. CNNs and Transformer can be combined in a parallel manner, which is formalized as follows:

$$\mathbf{P}_u^{CNNs} = \text{CNNs}(\text{Embed}(S_u)) \tag{1}$$

$$\mathbf{P}_u^{Tr} = \text{Transformer}(\text{Embed}(S_u)) \tag{2}$$

$$\mathbf{P}_u^{final} = \text{Fuse}(\mathbf{P}_u^{CNNs}, \mathbf{P}_u^{Tr}) \tag{3}$$

The final preferences $\mathbf{P}_u^{final}$ can be fed into a prediction module for recommending personalized items for user u. Meanwhile, CNNs and Transformer can be integrated in a cascaded manner, which is formalized as follows:

$$\mathbf{P}_u^{final} = \text{Transformer}(\text{CNNs}(\text{Embed}(S_u))) \tag{4}$$

AdaMCT [73] captures long-term and short-term dependencies by incorporating locality inductive bias into the Transformer with a local convolutional filter. CASFE [74] first leverages CNNs to capture periodic information, and then leverages Transformer to evaluate the importance of different behaviors for target behaviors.

4.4.2 Combination of GNNs and Transformer

GNNs and Transformer excel at modeling graph structure and sequences, respectively. Transformer can capture temporal patterns and long-term dependencies. Meanwhile, GNNs can capture structural information and users' long-term preferences. Therefore, combining GNNs and Transformer enables SR models to achieve better performance. GNNs and Transformer can be combined in a parallel manner, which is formalized as follows:

$$\mathbf{P}_u^{GNNs} = \text{GNNs}(\text{Embed}(\mathcal{G}), u) \tag{5}$$

$$\mathbf{P}_u^{Tr} = \text{Transformer}(\text{Embed}(S_u)) \tag{6}$$

$$\mathbf{P}_u^{final} = \text{Fuse}(\mathbf{P}_u^{GNNs}, \mathbf{P}_u^{Tr}) \tag{7}$$

In the meantime, GNNs and Transformer can be integrated in a cascade manner, which is formalized as follows:

$$\mathbf{P}_u^{GNNs} = \text{GNNs}(\text{Embed}(\mathcal{G}), u) \tag{8}$$

$$\mathbf{P}_u^{Tr} = \text{Transformer}(\text{GNNs}(\text{Embed}(\mathcal{G}), S_u)) \tag{9}$$

$$\mathbf{P}_u^{final} = \text{Fuse}(\mathbf{P}_u^{GNNs}, \mathbf{P}_u^{Tr}) \tag{10}$$

Compared with the former manner, it is noted that the latter leverages GNNs to extract item embeddings as the input of Transformer.

metaCSR [75] leverages Meta Learner to extract and propagate transferable knowledge from prior users and effectively learn presentations for cold-start users. GL-GraphFormers [76] leverages a global user-item bipartite graph to learn both first- and second-order graph information, and then injects the information into Transformer in the form of input and attention weights. MAERec [77] adaptively and dynamically distills sequential transition patterns for constructing a contrastive learning framework. SCG-SPRe [78] leverages a complement graph and a substitute graph to encode the complementary and substitutable relations, respectively. To explore global item-to-item interactions (collaborative information), APGL4SR [79] constructs a global item graph. Meanwhile, to capture personalized patterns, user embeddings are leveraged. MRGSRec [80] leverages a sequential encoder and a graph encoder to learn local behavioral representations and global behavioral representations, respectively. Subsequently, the two preferences are fused together for final recommendation. However, these methods have some drawbacks. For example, learning collaborative signals through graph neural networks is computationally expensive.

4.4.3 Other Combinations

RANN [81] leverages the strengths of RNNs and the self-attention mechanism in Transformer to capture user preferences from different views. RCNN [82] combines the advantages of RNNs and CNNs, in which RNNs can capture long-term dependencies and CNNs can capture short-term sequential patterns. NISRec [83] captures and aggregates the long-term and short-term intentions of neighbor users. Then NISRec [83] fuses these intentions for recommendation. Mixture networks can leverage multiple experts to learn diverse interests. For example, M3 [84] leverages a mixture of experts, where each expert specializes in capturing a specific temporal range. These experts are dynamically combined through a learned gating mechanism. Therefore, M3 [84] can capture diverse behavior patterns effectively.

By combining different techniques, SR models can achieve better performance. However, these models have several drawbacks. Firstly, they are more complex and computationally expensive. Secondly, different techniques might capture conflicting preferences. For example, Transformer and GNNs capture distinct preferences, which may conflict. Thirdly, different techniques converge at different rates.

5 SR with Side Information

In this section, our survey first discusses SR models combining IDs and one of the general features (i.e., categorical features and numerical features). Then, our survey introduces SR models combining IDs and reviews. Next, our survey shows SR models combining IDs and one of the graph-structure features (i.e., knowledge graph and social networks). Finally, our survey shows SR models combining IDs and multiple feature types.

Table 2 The advantages and disadvantages of different SR models.

SR models	Advantages	Disadvantages
Traditional SR models	Simple, can capture short-term sequential dependencies, efficient, good explainability	cannot capture long-term sequential dependencies, cannot handle complex data
RNNs-based SR models	Can capture long-term sequential dependencies	Are sensitive to the order of items in an interaction sequence
CNNs-based SR models	Can model union-level sequential patterns, are not sensitive to the order of interacted items	High computational costs due to lots of convolutional filters
Transformer-based SR models	Can identify the importance of interacted items, can capture short-term and long-term dependencies	Time complexity is related to the length of interaction sequences
Diffusion-based SR models	Can alleviate data sparsity, can generate diversity interaction data	Complex training process, high computational costs
GNNs-based SR models	Can model complex transitions among interacted items, can capture structure characteristics and collaborative signals from a global view	High computational costs, over-smoothing problem
MLPs-based SR models	Computational efficiency, a linear time and space complexity, simple model architecture	Limited performance on ultra-long sequences, weak in capturing sequential dependencies
VAE-based SR models	Can capture complex latent relationships, can handle the uncertainty of users' preferences	High computational costs, are influenced easily by potential noise
TCNs-based SR models	Can preserve sequence order during convolution process and conduct efficient sequence modeling	Are difficult to capture long-term sequential dependencies, high complexity for hyperparameters searching
Capsule-networks-based SR models	Can model multi-facet interests, can model complex interaction sequences	Are difficult to capture long-term sequential dependencies
RL-based SR models	Can adapt to changing sequence lengths, can model interaction process	cannot solve the cold-start problem, rely heavily on real-time feedback

5.1 One Categorical or Numerical Feature

Many SR models combine either a categorical or a numerical feature with item IDs to recommend items for users. Some specific methods are shown in Figure 7. From Figure 7, we can know that these methods can be grouped into four main types. The first method leverages two backbones to learn two kinds of preferences from one sequence containing item IDs and the other sequence containing feature k, respectively. These preferences are then fused using methods such as contrastive learning, concatenation, addition, gate addition, or cross-attention. The fusion process is formalized as follows:

$$\mathbf{P}_u^{final} = \text{Fuse}(\text{BM}_1(\mathbf{E}_u), \text{BM}_2(\mathbf{A}_u^{(k)})) \qquad (11)$$

The second method leverages the value of feature k to split the original sequence containing item IDs into sub-sequences. For example, some SR models split hybrid interaction sequences into separate interaction sequences in each domain [85, 86]. Different preferences are then extracted from these sub-sequences and fused together to predict personalized items. The fusion process is formalized below:

$$\begin{aligned}\mathbf{P}_u^{final} = \text{Fuse}(&\text{BM}_1(\text{Embed}(S_{u,1}^{(k)})), \ldots, \\ &\text{BM}_m(\text{Embed}(S_{u,m}^{(k)})))\end{aligned} \tag{12}$$

It is noted that the fusion usually exhibits a certain directionality, where user preferences are transferred from a source domain to a target domain [87]. The third method leverages simple techniques to combine item embeddings and feature embeddings. Then, the fused embeddings are fed to a backbone model for recommendation. For example, adding them together is to obtain the input of a backbone model. The fusion process is formalized as follows:

$$\mathbf{P}_u^{final} = \text{BM}_1(\text{Fuse}(\mathbf{E}_u, \mathbf{A}_u^{(k)})) \tag{13}$$

The last method integrates item embeddings with feature embeddings in a backbone model. For example, leveraging feature embeddings changes the weights in attention matrices. The fusion process is formalized as follows:

$$\mathbf{P}_u^{final} = \text{BM}_1(\mathbf{E}_u, \mathbf{A}_u^{(k)}) \tag{14}$$

Though all four kinds of methods can obtain final user u's preferences, they vary in terms of complexity and efficiency [87, 88]. Meanwhile, the predicted goal might contain the feature of the next interacted item [25, 89].

5.1.1 Behaviors

Different behaviors show users' different preferences. Click behaviors usually reflect users' short-term preferences. Favorite behaviors usually reflect users' middle-term preferences. Purchase behaviors usually reflect users' long-term preferences. Meanwhile, dependencies exist among these different behaviors. MB-STR [90] injects users' behaviors into Transformer to capture multi-behavior heterogeneous dependencies. PBAT [91] models users' personalized behavior patterns and behavioral collaborations by a personalized behavior pattern generator and a behavior-aware collaboration extractor, respectively. GHTID [92] leverages a global heterogeneous graph (constructed from all interaction sequences) and local heterogeneous graphs (constructed from each interaction sequence) to explicitly learn heterogeneous item transitions.

5.1.2 Timestamp

Users usually exhibit certain behaviors at specific times. For example, users tend to purchase T-shirts in summer. By considering timestamps, SR models can capture users' evolving interests more precisely. DRL-SRe [93] exploits time-sliced graph neural networks to learn complex user-item interactions from a global perspective, and leverages GRUs to learn complex item-to-item correlations. To account for temporal context in recommendations, MOJITO [94] leverages a Gaussian mixture method to combine attention-based temporal context with item representations. TGSRec [95] leverages a temporal collaborative Transformer layer to capture collaborative signals and temporal dynamics. TIGSA [96] leverages a time interval-aware graph to consider the impact of time intervals for recommendation, and leverages Transformer to extract users' sequential preferences. TMGN [97] first identifies the k most similar users to each candidate user. And then, TMGN [97] injects multiple users' sequences and temporal information into multi-head attention. By leveraging timestamps, HyperRec [98] splits a whole user-item interaction hypergraph into sub-hypergraphs. HyperRec [98] adopts these sub-hypergraphs and multiple convolutional layers to capture multi-order connections

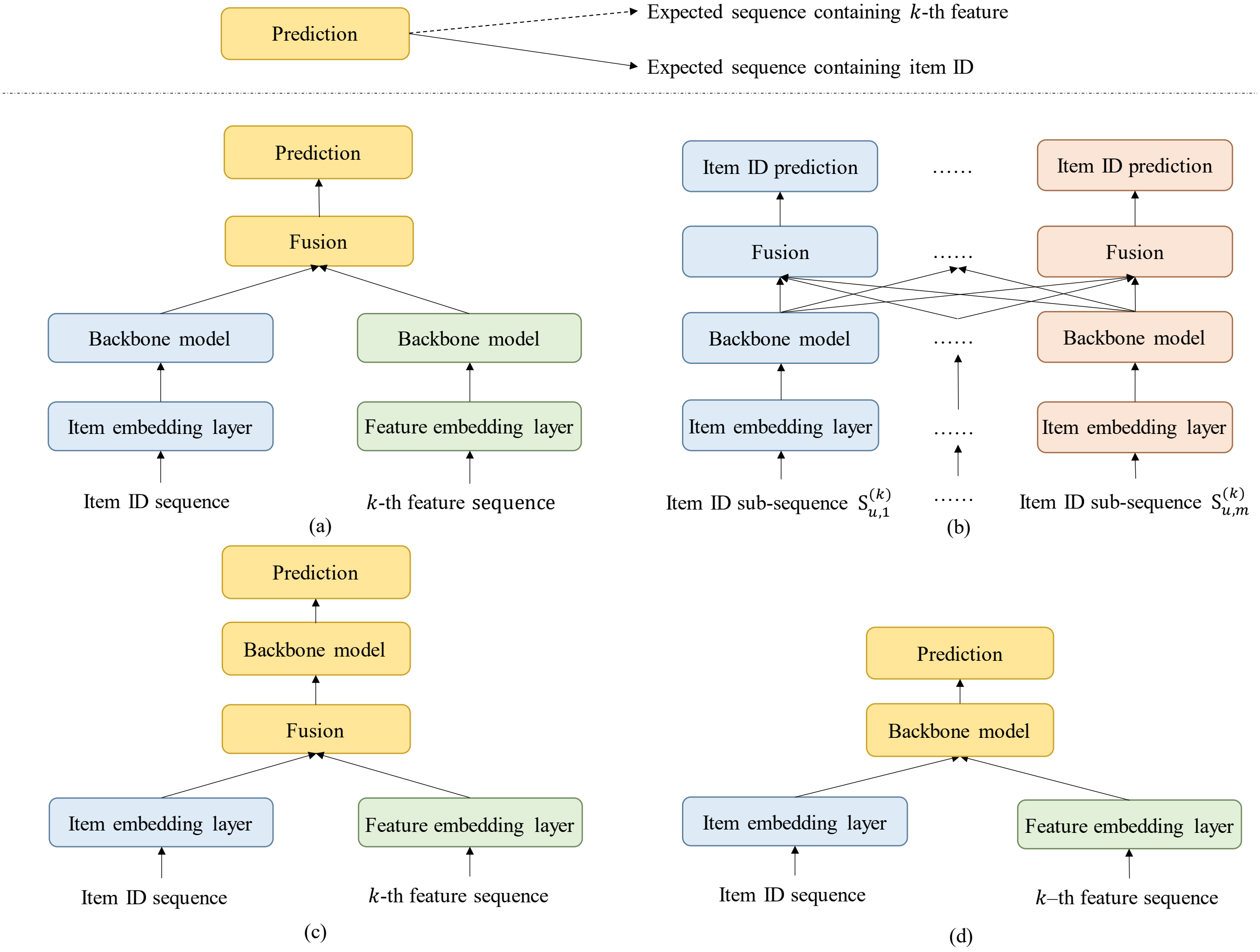

Fig. 7 Illustration of the usage of one categorical or one numerical feature in sequential recommendation.

and short-term user intents. Then the dynamic item embeddings and short-term user intents are incorporated by a fusion layer, and the incorporated representations are fed into a self-attention layer. PT-GCN [99] leverages a position-enhanced and time-aware graph convolution operation to capture sequential patterns and temporal dynamics.

5.1.3 Category

The category of interacted items can reflect users' preferences more precisely. HierTrans [100] leverages a novel hierarchical temporal graph to extend traditional item-level relations to category-level relations. HPM [101] leverages Transformer to learn both low-level preferences (i.e., item IDs) and high-level preferences (i.e., item categories). While considering both item IDs and the categories of items, AutoMLP [102] captures users' long-term and short-term interests, with short-term interests learned by an adaptive search algorithm.

5.1.4 Domain

Domain information plays a crucial role in sequential recommender systems. SR models usually handle separate interaction sequences in each domain. Then SR models transfer the knowledge obtained from a source domain to a target domain [86, 103]. It is because data distribution is different across domains. C^2DSR [87] constructs a global item-to-item graph across two domains to learn transitions between two domains. DDGHM [104] constructs local graphs and a global graph to capture intra-domain sequential transitions and inter-domain sequential transitions, respectively. And, DDGHM leverages hybrid metric learning to alleviate data sparsity and improve recommendation performance.

5.1.5 Rating

Users usually rate interacted items, indicating their preferences. By considering ratings, users' preferences can be more accurately captured. TRA-SR [105] integrates rating information into the weight calculation of self-attention to enhance the learning of users' preferences.

5.1.6 Price

In real life, users often prefer purchasing cheaper items. Even if users have a strong liking for an item, they may not purchase it if it is too expensive. Both price and interest preferences might influence users' purchase choices simultaneously. Therefore, CoHHN [106] designs a co-guided heterogeneous hypergraph network to simultaneously extract users' price preferences and interest preferences. Then, CoHHN leverages a co-guided learning schema to model the complex relations between the two kinds of preferences. Finally, CoHHN predicts users' behaviors based on the two kinds of preferences and item features.

5.2 Graph Structure Features

5.2.1 Knowledge Graph

Knowledge graph is a structure representation of knowledge, where entities (nodes) are connected by relationships (edges). The entities can represent users, items and other related features. The relationships represent meaningful connections (e.g., "a user purchases a book" and "The color of the iPhone is black"). Therefore, knowledge graph can describe the relations among items, users, and features, as well as interactions between users and items. Knowledge graph provides rich extra knowledge for recommending personalized items. Therefore, by leveraging knowledge graph, SR models can alleviate cold-start and data sparsity problems. Meanwhile, SR models based on knowledge graph are also capable of providing explanations for recommendation.

There are several SR models based on knowledge graph. EIUM [107] leverages knowledge graph to conduct explainable recommendation. KV-MN [108] integrates knowledge graph into RNNs to enhance semantic representations. KP4SR [109] leverages knowledge graph to capture users' fine-grained preferences. Meanwhile, KP4SR [109] converts knowledge graph to knowledge instructions for noise reduction. KERL [110] incorporates knowledge graph into reinforcement learning. In order to learn sequential dependencies while considering semantic information, KERL [110] leverages a composite reward function to compute both sequence-level and knowledge-level rewards. TKG-SRec [111] leverages temporal knowledge graph to capture temporal information, and leverages GRUs to conduct temporal knowledge evolution training. However, SR models based on knowledge graph have several disadvantages as follows: 1) Complexity: Knowledge graph contains many entities and relationships, which may increase the complexity of SR models and cost more computational resources. 2) Noise: Constructing a high-quality knowledge graph is challenging. Incorrect relationships within the knowledge graph might degrade the recommendation performance.

5.2.2 Social Network

Social networks provide insight into the friend relationships among users. Since users tend to have similar preferences to their friends, leveraging social networks allows SR models to capture users' preferences more accurately. Satori [112] leverages a graph attention network and a self-attention mechanism to extract auxiliary features and user intentions, respectively. Then, hybrid user and item representations are sent to a prediction module to calculate predicted scores. SERec [113] first constructs a heterogeneous graph by combining a social network and users' interaction sequences. Then by leveraging GNNs, SERec improves user and item representations by integrating knowledge from the social network.

5.3 Review

Compared with rating information, reviews can reflect users' preferences about items comprehensively in the form of text and images. Meanwhile, reviews can also provide more detailed information about items. By reading reviews given by other people, users' action decisions might be influenced. RNS [114] learns users' long-term preferences by an aspect-aware convolutional network over users' document. Meanwhile, RNS learns users' short-term preferences by hierarchical attention. Finally, RNS combines the two kinds of preferences for recommendation. To consider both user-item interactions and reviews simultaneously, PIRSP [115] learns item sequential patterns and review sequential patterns together. Then PIRSP introduces a fusion gating mechanism to learn short-term preferences from the two patterns. Finally, PIRSP combines short-term and long-term preferences for recommendation. CCA [116] designs a cascaded cross-attention mechanism to capture the complex relationships among an item sequence, a review sequence and candidate items.

5.4 Fusion of Different Features

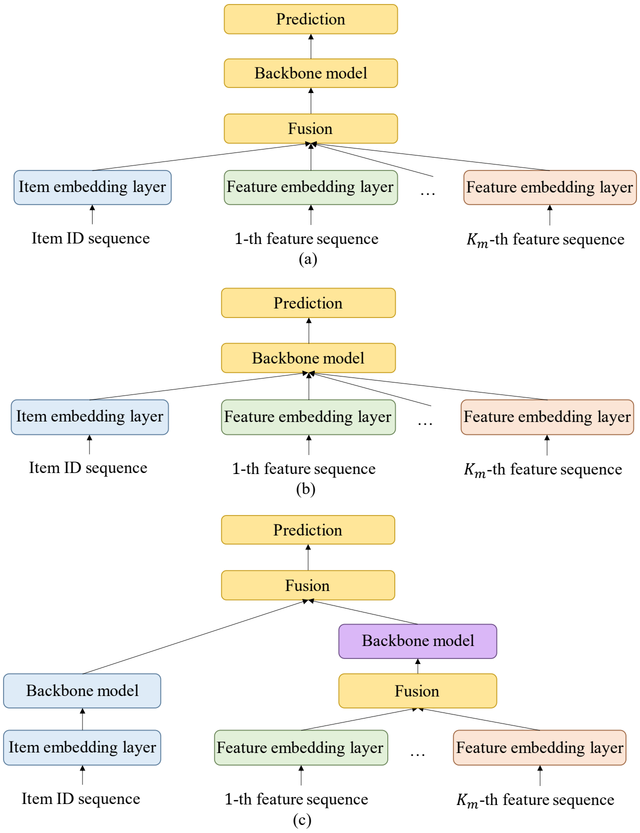

Fig. 8 Illustration of different fusion strategies for multiple general features.

In SR models, multiple features are usually used. Different features have different fusion methods. These fusion methods can mainly be summarized into three types. Some specific details are shown in Figure 8. The first type leverages simple methods to fuse item embeddings and feature embeddings together, for example, adding them together is to obtain the input of a backbone model. The fusion is shown as follows:

$$\mathbf{P}_u^{final} = \mathrm{BM}_1(\mathrm{Fuse}(\mathbf{E}_u, \mathbf{A}_u^{(1)}, \ldots, \mathbf{A}_u^{(K_g)})) \tag{15}$$

The second type fuses different features in a backbone model [25], for example, leveraging a decoupled features fusion mechanism in a self-attention mechanism. The fusion is shown as follows:

$$\mathbf{P}_u^{final} = \mathrm{BM}_1(\mathbf{E}_u, \mathbf{A}_u^{(1)}, \ldots, \mathbf{A}_u^{(K_g)}) \tag{16}$$

The third type of methods first fuses different feature embeddings together, and then passes the fused embeddings through a backbone model to learn feature representations. Meanwhile, item representations are learned by the other backbone model. Finally, the third method fuses item representations and feature representations together for prediction. The fusion is described as follows:

$$\mathbf{P}_u^{feat} = \mathrm{BM}_1(\mathrm{Fuse}(\mathbf{A}_u^{(1)}, \ldots, \mathbf{A}_u^{(K_g)})) \tag{17}$$

$$\mathbf{P}_u^{final} = \mathrm{Fuse}(\mathrm{BM}_2(\mathbf{E}_u), \mathbf{P}_u^{feat}) \tag{18}$$

5.4.1 Common Features Fusion

In this section, we mainly introduce some SR models that fuse some common features (i.e., position, category, brand ,etc.) and item IDs. The structure of these SR models is features-agnostic to some extent. ICAI-SR [3] leverages a heterogeneous graph to capture complex item-to-attribute relationships by inner attribute aggregation and attribute-to-item aggregation, and then passes aggregation embeddings through an entity sequential model. MLP4Rec [117] designs a fusion mechanism to capture sequential, cross-channel and cross-feature correlations. SMLP4Rec [118] leverages a layer normalization for a sequence mixing module, a feature mixing module and a channel mixing module simultaneously. Meanwhile, to enhance efficiency and effectiveness, SMLP4Rec leverages a parallel mode. RetaGNN [119] leverages a relational attentive GNN in which learnable weight matrices focus on various relations among users, items and attributes to have inductive and transferable properties. Then, RetaGNN [119] leverages Transformer to capture short-term and long-term temporal patterns.

In Transformer-based SR models, several advancements have been made in integrating common features with item IDs. FDSA [120] first fuses different features by a vanilla attention mechanism. Then, FDSA leverages separate self-attention blocks to independently learn feature-level and item-level representations. Finally, FDSA fuses the two kinds of preferences for recommendation. SASRec$_F$ [25] is a sample model to fuse common feature embeddings and item ID embeddings through a concatenation operation. This fusion method is considered invasive [4], similar to an addition operation and a gating addition operation. The specific process is shown in part (a) of Figure 8. This fusion method might compromise the original information in item embeddings, potentially leading to negative effects. To address this issue, NOVA [4] leverages a non-invasive method to enhance the learning of attention matrices. In NOVA, key and query matrices are derived by integrating different feature embeddings and item embeddings while value matrices are directly obtained from item embeddings. However, NOVA [4] still leverages integrated embeddings to learn key and value matrices. The compounded embedding space might introduce random disturbances and share identical gradients, which might degrade recommendation performance. DIF-SR [25] leverages decoupled attention to adaptively learn different features by flexible gradients. MSSR [121] captures the associations among an item ID sequence and each feature sequence. ASIF [122] can maintain semantic consistency between item IDs and features. Meanwhile, ASIF [122] can make full use of different kinds of features without compromising item embeddings, thanks to contrastive learning.

5.4.2 Context Information Fusion

Context information can reflect the specific environment where behaviors happen. SR models can provide more accurate recommendations by considering the context information. Bi-CASER [123] uses a context graph to represent context information, and then leverages bi-Transformer to learn users' sequential behaviors. CRNNs [124] incorporates context information (e.g., interaction timestamps) by combining item embeddings with context embeddings to recommend personalized items. STAR [125] leverages stacked RNNs to model the sequential and context-aware information simultaneously. STEM [126] considers item behaviors and leverages Transformer-based architecture for exploiting spatial and temporal information.

5.4.3 Special Features Fusion

In this section, we mainly introduce SR models designed to fuse special features. These SR models are typically applied to situations where certain special features (e.g., social networks and timestamps) are available. STEN [26] introduces temporal information and a social network to capture users' fine-grained dynamic interests in an event-level direct paradigm. DUVRec [127] constructs a timespan-aware sequence graph and an attribute-augmented graph to learn item-view and factor-view user representations, respectively. Recommendation performance is then improved by fusing the two representations. Based on features and labels, STARec [128] retrieves a target user's historical behaviors through a search-based retriever. Then, STARec [128] leverages graph structure to enhance representation learning. SGCross [129] transfers knowledge from an auxiliary domain to a target domain from a personal view, a temporal view and a collaborative view. The knowledge encompasses personal preferences, dynamic preferences and collaborative preferences.

6 Recent SR Advancements

In this section, we will introduce recent research directions in sequential recommendation (SR) models. Firstly, we will discuss SR models using multi-modal features. Secondly, we will explore a novel paradigm (i.e., generative recommendation) in sequential recommendation. Thirdly, we will introduce SR models using LLMs. Fourthly, we will introduce SR models designed to address ultra-long interaction sequences. Finally, we will present augmentation techniques aimed at enhancing SR performance.

6.1 Multi-Modal SR

A user u can be represented by the user's profile. The profile may consist of user u's demographic information or user u's interacted items. An item can be denoted by an item ID embedding or a pre-trained modality embedding. Modality content features consist of text, images, videos, audio, text-image multi-modal pairs, and so on. ID-agnostic representations can be learned using pre-trained

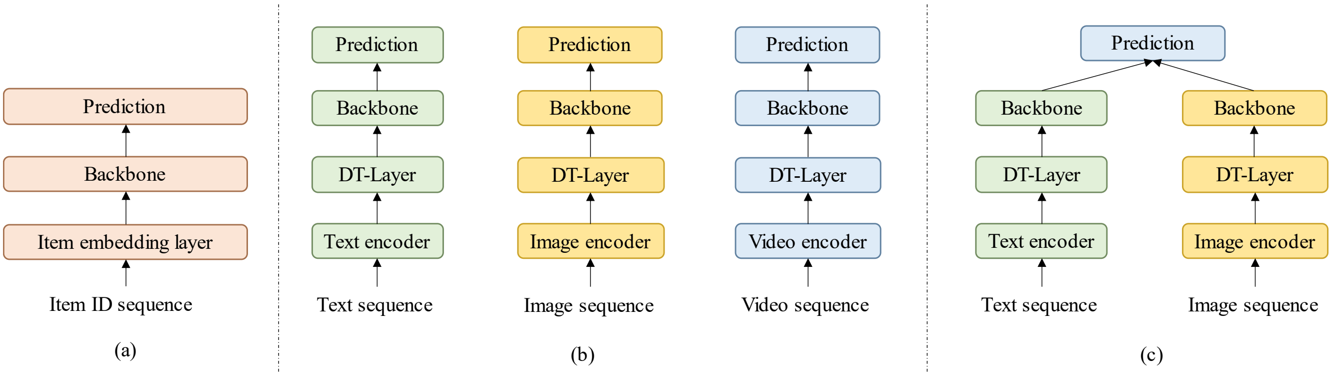

Fig. 9 Illustration of different modal-based SR models. Note that DT denotes the dimension Transformer.

modality models. A common text encoder is BERT, which is usually used to extract semantic features from text. Swin Transformer is an image encoder, which can be leveraged to extract features from images. The modal-based SR models are shown in Figure 9. Different from ID-based SR models, one type of modal-based SR models leverages a single modality feature (e.g., text or images) to leverage items. The other type of modal-based SR models leverages some modal features (e.g., text and images) to recommend personalized items [130]. In order words, the representations of item i are obtained from the multi-modal features of item i in modal-based SR models, which is formalized as follows:

$$\mathbf{e}_i = \begin{cases} \text{TextEncoder}(\text{Text}_i) \\ \text{ImageEncoder}(\text{Image}_i) \\ \text{VideoEncoder}(\text{Video}_i) \end{cases} \tag{19}$$

SR problems can be solved by NLP models. Different from natural language, interacted items do not typically follow a specific order in user interactions. If we swap two items in a user's interaction sequence, the user's preferences might not be changed. If we swap two words in a sentence, the meaning of this sentence will be changed. Text contains rich semantic information in the real world. By using natural language like text, semantic gaps across different domains and platforms can be bridged. Therefore, SR models can be trained in a universal semantic space.

UniSRec [8] leverages an MoE-enhanced adaptor and parametric whitening for domain adaptation. To learn universal sequence representations better, UniSRec proposes a seq-item contrastive task and a seq-to-seq contrastive task. The model is pre-trained in one domain and subsequently fine-tuned in another domain using either an inductive or transductive approach. Meanwhile, UniSRec adopts two augmentation strategies (i.e., item drop and word drop). VQ-Rec [27] is a novel approach to learn vector-quantized item representations for solving the problem about over-emphasizing the effect of text features, which might lead to poor recommendation performance. VQ-Rec converts item text to a series of codes. Then with a lookup over a code embedding table, VQ-Rec can obtain item embeddings. Based on representation schema, VQ-Rec leverages contrastive learning among the next ground-truth items and negative items (i.e., semi-synthetic negative items and mixed-domain negative items). Finally, based on the code-embedding method, VQ-Rec implements a novel cross-domain fine-tuning method. TransRec [131] is pre-trained in one domain, and then is fine-tuned in another domain. The TransRec [131] achieves excellent recommendation performance. It is because that TransRec [131] first obtains knowledge from the source domain, and then transfers the obtained knowledge to the target domain. The three models mentioned above share a similar approach: they are initially pre-trained in one domain and then fine-tuned in another domain. IDA-SR [132] adopts a pre-trained text encoder to learn item representations by encoding item text descriptions. The pre-training tasks consist of next-item prediction, masked item prediction and permuted item prediction. These item representations can then be used for personalized recommendations through semantic transfer. SASRec$_V$ [133] leverages videos to denote an item. Compared with SASRec using an item ID to denote an item, SASRec$_V$ [133] can achieve comparable performance using an end-to-end framework. RECFORMER [134] mainly focuses on embedding initialization. By adopting behavior-tuned pre-trained language models for the initialization of ID-based SR models, RECFORMER [134] can achieve better recommendation performance without introducing additional computational costs. SRT-1K [135] calculates item embeddings using a trainable feature extractor, which can make the number of parameters be independent of the number of items. Additionally, SRT-1K [135] leverages contrastive learning to enhance catalog diversity representation. RECFORMER [136] flattens item attributes into sentences to represent items. The SR models mentioned above achieve improved recommendation performance by combining item IDs and multimodal features.

ODMT [130] fuses different modal information (i.e., text and images) together. Meanwhile, ODMT uses knowledge distillation to model weights for online recommendations. MMSR [121] integrates multi-modal information into nodes. MMMLP [137] incorporates multi-modal data into MLPs to enhance recommendation performance. p-RNN [138] leverages four different frameworks to combine some item features (i.e., images and text) and item IDs for modeling interaction sequences. Based on the aforementioned modal-based SR models, the following conclusions can be drawn:

- Modal-based SR models with an end-to-end training framework achieve comparable recommendation performance with pure ID-based SR models. Because the pre-trained features might not be suitable for recommendation.
- Modal-based SR models with larger parameters tend to have better recommendation performance than the counterpart containing smaller parameters in recommendation tasks.
- In a cold-start setting, modal-based SR models can obtain better recommendation performance than pure ID-based SR models.
- If we leverage a pre-trained modality model to extract features, we can obtain better performance than than training a model from scratch.

In the meantime, the modal-based SR models have some advantages as follows:

- Leveraging modality information to denote an item has the advantages of interpretability and visualization.
- Modal-based SR models can alleviate the cold-start problem.
- Modal-based SR models can realize the transferability of the models across different platforms.
- Multi-modal features act as a bridge to better integrate recommendation and search algorithms.

The modal-based SR models also have several disadvantages as follows:

- Modal-based SR models require substantial computational resources and extended training time.
- The hyper-parameters need to be searched carefully, as improper hyper-parameters easily lead to model collapse.
- Semantic information from different modalities might not be suited for recommendation.

6.2 Generative Recommendation

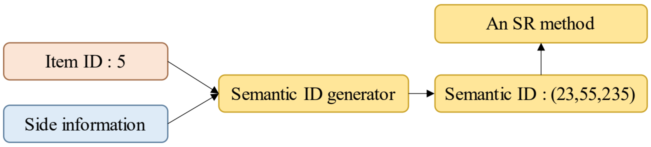

Fig. 10 Illustration of semantic ID generation.

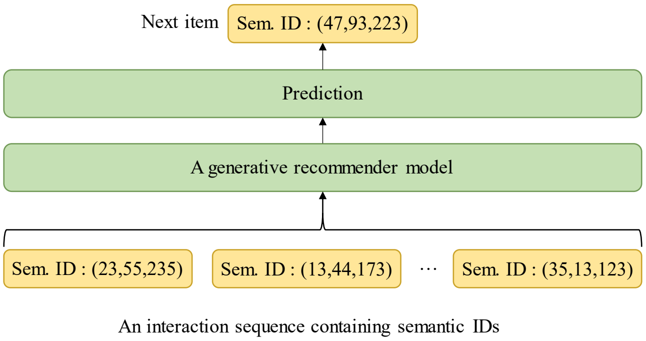

Fig. 11 Illustration of generative recommendation with semantic IDs.

Generative recommendation is proposed on the basis of a widely used technique in document retrieval (i.e., generative retrieval) [139]. Different from some above-mentioned SR models, generative recommendation models [139, 140] leverage semantic IDs to denote an item. As depicted in Figure 10, a semantic ID consists of a series of tokens. Compared with only using an item ID to denote an item, this method has several advantages as follows:

- Compared with item IDs, semantic IDs encode semantic information extracted from side information (e.g., text descriptions) more efficiently.
- Because the semantic IDs of different items might contain some of the same tokens, knowledge can be shared among similar items.
- Using semantic IDs prevents the embedding table size from increasing linearly with the number of items in datasets. It is because an item is denoted by a series of tokens instead of a unique item ID. The total number of tokens is far smaller than the number of all item IDs.

Some above-mentioned methods leverage the multimodal features of an item to denote the item. However, these methods might damage fine-grained semantic information [141]. Semantic IDs can solve this problem. Due to their hierarchical nature, semantic IDs contain semantic information that ranges from a coarse-grained level to a fine-grained level. Generative recommendation generally involves: semantic ID generation and generative recommendation with semantic IDs. As depicted in Figure 10, semantic IDs are obtained by encoding item IDs and side information about items. RQ-VAE is a widely-used semantic ID generator, which leverages a residual quantizer to learn quantized representations [139]. Because target items are denoted by semantic IDs, a generative recommender model need to generate the target items by an auto-regressive manner. A widely-used generative recommender model is based on a Transformer-based encoder-decoder architecture [142, 143]. The process is shown in Figure 11. In generative recommender models, the probability of user u interacts with item i is calculated by chain rule, which is formalized as follows:

$$P(i|S_u) = \prod_{j=1}^{l} p\left(y_{i,j}|S_u, y_{i,1}, y_{i,2}, \ldots, y_{i,j-1}\right) \quad (20)$$

In recent years, some generative sequential recommender models have been proposed. TIGER [139] is the first model that leverages semantic IDs for generative recommendation. MBGen [140] considers the role of behavioral information in generative recommendation. To consider the complementary nature between behavioral information and semantic information, EAGER [142] leverages a shared encoder and two separate decoders to decode the two kinds of information. CoST [143] leverages a contrastive quantization-based semantic tokenization approach to capture neighborhood relationships among items. LETTER [141] leverages semantic, collaborative and diversity regularization. The first two types of regularization can extract semantic information and collaborative signals, respectively. The diversity regularization can mitigate code assignment bias. To consider the multi-modal features of items, MMGRec [144] leverages a hierarchical quantization method (i.e., Graph RQ-VAE) by incorporating multi-modal features. Compared with traditional SR models, generative recommender models have some advantages as follows:

- Generative recommender models have strong capabilities in recommending unpopular items. Therefore, they can alleviate the cold-start problem.

- By leveraging temperature-based sampling, generative recommender models can generate diverse recommendation lists.

However, generative recommender models also have some disadvantages. For example, they might generate invalid semantic IDs during the prediction phase.

6.3 LLM-powered SR

6.3.1 Large Language Models

Large language models (LLMs) can handle more complex tasks compared with traditional language models based on deep learning. The main characteristics of LLMs are their billions of parameters as well as the vast amounts of data and computational resources used during training. Therefore, LLMs have a deeper understanding of natural language and extensive knowledge extracted from amounts of data. There are many kinds of LLMs. In our survey, we introduce them briefly as follows:

- LLaMA: LLaMA [4] (Large Language Model Meta AI) is developed by Meta. It is a family of large language models, including LLaMa1, LLaMa2 and LLaMa3.
- ChatGLM: ChatGLM (Generative Language Model) is developed by Tsinghua University, which is a bilingual model for understanding and generating both Chinese and English.
- T5: T5 (Text-To-Text Transfer Transformer) is developed by Google, which can solve many kinds of tasks by modeling each task as a text-to-text task.
- ChatGPT: ChatGPT [5] is the most popular large language model, which is developed by OpenAI. It includes models such as GPT-3.5, GPT-4, GPT-4o, and other variants.
- PaLM: PaLM (Pathways Language Model) is developed by Google, which is one of the largest language models with up to about 540 billion parameters. It shows strong capabilities in multilingual tasks and source code generation.

[4] https://huggingface.co/meta-llama/Llama-2-13b

[5] https://platform.openai.com/

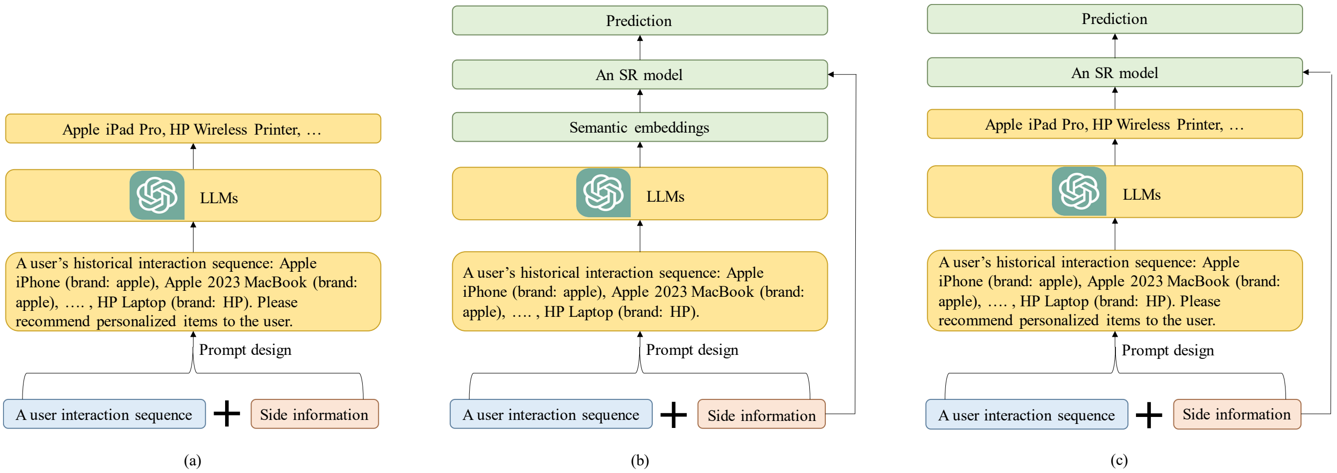
Apple iPad Pro, HP Wireless Printer, …
LLMs
A user's historical interaction sequence: Apple iPhone (brand: apple), Apple 2023 MacBook (brand: apple), …. , HP Laptop (brand: HP). Please recommend personalized items to the user.
Prompt design
A user interaction sequence
+
Side information
(a)
Prediction
An SR model
Semantic embeddings
LLMs
A user's historical interaction sequence: Apple iPhone (brand: apple), Apple 2023 MacBook (brand: apple), …. , HP Laptop (brand: HP).
Prompt design
A user interaction sequence
+
Side information
(b)
Prediction
An SR model
Apple iPad Pro, HP Wireless Printer, …
LLMs
A user's historical interaction sequence: Apple iPhone (brand: apple), Apple 2023 MacBook (brand: apple), …. , HP Laptop (brand: HP). Please recommend personalized items to the user.
Prompt design
A user interaction sequence
+
Side information
(c)

Fig. 12 The common SR models based on LLMs.

- Vicuna: Vicuna [6)] is designed by fine-tuning LLaMA on high-quality conversational data.
- Qwen: Qwen is developed by Alibaba and excels at processing some tasks described in both Chinese and English.

Large language models have a strong ability to understand text descriptions and integrate external knowledge. Consequently, incorporating LLMs into sequential recommendation (SR) models can enhance performance and alleviate challenges such as cold-start and data sparsity problems. As shown in Table 3, there are some SR models leveraging LLMs for recommendation. The specific details are shown as follows:

6.3.2 LLMs for Recommendation

There are several models leveraging LLMs for recommendation. Some specific details are shown in Figure 12. The first type of models leverages LLMs to recommend items to users directly, which is formalized below:

$$\hat{x}_{prompt} = \text{LLMs}(x_{prompt}) \tag{21}$$

By reading the outputs of LLMs $\hat{x}_{prompt}$, we can identify the names of recommended items.

The second type utilizes semantic embeddings extracted from LLMs to recommend personalized items. As shown in part (b) of Figure 12, the prompt x_{prompt} might contain the text descriptions about a user's whole historical interaction sequence. Similar to the modal-based SR models, the prompt x_{prompt} might be the text descriptions of item i, and LLMs are leveraged to extract semantic information about item i, which is formalized as follows:

$$\mathbf{e}_i = \text{LLMs}(x_{prompt}) \tag{22}$$

Compared with conventional text feature extractors such as BERT, LLMs can extract richer semantic information.

The third type of models leverages LLMs to generate high-quality data which is then combined with the original data to train SR models. The data generation process is formalized as follows:

$$\hat{x}_{prompt} = \text{LLMs}(x_{prompt}) \tag{23}$$

Compared with the above-mentioned first type, the outputs of LLMs are regarded as augmented data for training. It is noted that the x_{prompt} might vary with the usage of LLMs changing.

[6)] https://github.com/lm-sys/FastChat

Table 3 The SR models based on LLMs.

Large Language Models	Representative Works
T5	[145], [146]
LLaMA	[147], [148], [149], [150], [151], [152], [153], [154], [155], [9]
ChatGLM	[156]
ChatGPT	[157], [158], [159]
Vicuna	[160], [161]
Qwen	[162]

The most straightforward approach is to directly leverage LLMs for recommending personalized items. Given user u's historical sequence and a target item, LLMs predict u's preferences for the target item. Meanwhile, given the interacted item sequences, LLMs can generate corresponding recommendation lists. For example, P5 [145] can generate personalized recommend items by inputting historical interaction items into LLMs. There are different tasks, such as generating an item or a list of items, rating items, and ranking the order of the interacted items. GenRec [9] leverages an adaptation method (LoRA) to fine-tune and perform inference tasks on LLaMA. However, generative recommendation models have some weaknesses, such as the potential to generate non-existent items.

6.3.3 LLMs for Learning Semantic Embeddings

Some SR models use LLMs to initialize item embeddings. Firstly, RecInterpreter [149] leverages a linear layer to map item embeddings from an item ID embedding space to a text embedding space. By providing task-specific textual prompts and projected hidden representations, LLaMA is encouraged to generate textual descriptions of items in interaction sequences. The second task aims to discover residual items by task-specific textual prompts and hidden representations. To solve the problem of long-tail items, LLM-ESR [157] leverages LLMs to generate semantic embeddings for enhancing the semantic information of items. Common methods are to combine item ID embeddings and generated semantic embeddings for achieving better recommendation performance. To combine the advantages of traditional SR models and LLMs, E4SRec [152] uses item IDs as input. This ensures that all the generated recommendation items fall within candidate lists. Therefore, E4SRec can handle item IDs more effectively. Though knowledge can be transferred from some source domains to a target domain, the cold-start problem cannot be completely solved. LLMs cannot capture collaborative information effectively. Traditional cross-domain sequential recommendation models can capture collaborative signals while LLMs-based models excel at capturing semantic information. URLLM [153] incorporates both advantages. Firstly, URLLM leverages a dual-graph neural network to capture collaborative signals and transfer knowledge from a source domain to a target domain. Meanwhile, URLLM leverages a retrieve-generation model to incorporate structure information into an LLMs-based recommender model. Finally, URLLM combines collaborative information and semantic information to recommend items. From these models, we can draw the following conclusions: 1) Initializing item embeddings with a large language model can obtain recommendation performance gains. 2) Fine-tuning LLMs for recommendation tasks enables LLMs to learn domain-specific and domain-shared knowledge.

To reduce computational overload and enhance the efficiency and effectiveness of inference, Lite-LLM4Rec [146] leverages a beam search decoding method and a hierarchical LLM structure for recommendation. We can leverage prompts to conduct fine-tuning of LLMs. TALLRec [150] employs two tuning methods, such as Alpaca tuning and rec-tuning, to optimize the tuning process of LLMs more efficiently. SeRALM [154] leverages item IDs and knowledge generated by LLMs as input. Meanwhile, SeRALM leverages the feedback from recommender systems to refine the knowledge of items generated by LLMs. To extract semantic information generated by LLMs and enhance computational efficiency, SAID [155] leverages a projector to map item IDs into item embeddings and feeds them into LLMs for refining the item embeddings. Finally, SAID leverages the trained projector to recommend items more effectively.

Knowledge distillation and contrastive learning are effective techniques for extracting knowledge from LLMs. Knowledge distillation can enhance the efficiency of LLMs. SLIM [147] can obtain semantic knowledge from LLMs through knowledge distillation. DLLM2Rec [151] distills the knowledge from LLMs-based SR models to conventional SR models by leveraging two kinds of distillation (i.e., collaborative embedding distillation and importance-aware ranking distillation). LSVCR [156] is trained in two stages (i.e., personalized preferences alignment and recommendation-oriented fine-tuning). It uses contrastive learning between preferences generated by LLMs-based recommendation models and preferences generated by conventional SR models. Some conventional SR models can obtain richer semantic information.

LLMs often struggle with processing long text descriptions. LLM-TRSR [148] can leverage too-long text descriptions as the input of LLMs by segmenting original users' interaction sequences. The LLMs-based SR models will consume lots of computational resources when processing long interacted sequences. To solve this problem, BAHE [162] leverages a behavior-aggregated hierarchical encoding method to enhance computational efficiency. By encoding a user's separate interacted sequences in parallel, BAHE can obtain the user's final representations by concatenating them together. Because LoRA's shared parameters across all users limit its ability to recommend personalized items. Meanwhile, ultra-long interaction sequences might affect the efficiency of LLMs. RecLoRA [161] proposes a personalized LoRA module to overcome this limitation and designs a long-short modality retriever to retrieve interaction sequences of different lengths based on different modalities.

6.3.4 LLMs for Data Generation

LLMs can be leveraged to generate data for solving the data sparsity and cold-start problems. For example, KDA$_{LRD}$ [159] leverages LLMs to describe relationships among items. Most existing models rely on predefined relations among items, which might result in sparse issues. To generate more item-to-item relations, KDA$_{LRD}$ leverages LLMs to learn item representations. The item reconstruction module leverages the estimated relation and another item to reconstruct the target item. Finally, KDA$_{LRD}$ leverages the predefined and generated relations for recommendation. FineRec [158] initially leverages LLMs to extract attribute pairs from reviews. Then by constructing an attribute-aware graph and aggregating, FineRec can obtain improved user and item embeddings for recommendation. For few-shot recommendation, ReLLa [160] can be used as a data augmentation method to augment data.

ReLLa [160] leverages LLMs to retrieve the top-k semantically relevant behaviors according to a target item. Leveraging the semantic knowledge of LLM, ReLLa enhances the quality of data. Finally, ReLLa is trained on a mixed dataset containing the original data and the retrieval-enhanced data.

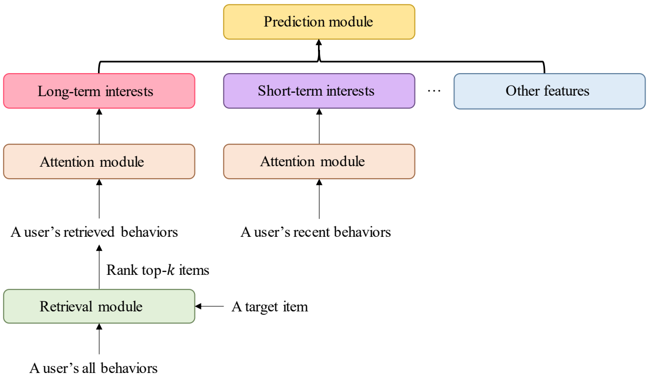

Fig. 13 Illustration of a typical ultra-long sequential recommendation method.

6.4 Ultra-Long SR

An ultra-long behavior sequence typically refers to a scenario where the number of interacted items exceeds 1000. Compared with conventional behavior sequence modeling in SR models, long behavior sequence modeling presents several challenges. Firstly, modeling long sequences requires substantial computational resources. Secondly, long sequences often contain noise due to irrelevant behaviors. Finally, during the inference phase, modeling long sequences can be computationally expensive. For example, the time complexity of Transformer is $O(L^2)$. Therefore, SR models while modeling long sequences might face high latency. The general long sequence modeling methods are shown in Figure 13. Firstly, the most relevant k items are retrieved from a whole interaction sequence. Then long-term interests are extracted from the retrieved k items by an attention module. Short-term interests are extracted from recently interacted items by an attention module. The process of extracting the two interests is formalized as follows:

$$\mathbf{P}_u^{short} = \mathrm{Att}(S_{u,recent}) \tag{24}$$

$$\mathbf{P}_u^{long} = \mathrm{Att}(\mathrm{Ret}(S_u)) \tag{25}$$

The sequences fed into the attention module are relatively short, so computational efficiency can be enhanced to some extent. Finally, by leveraging the long-term interests, the short-term interests and other features, SR models can recommend personalized items for users.

Traditional SR methods focus on leveraging the most recent behaviors, while our UBR4CTR [163] emphasizes the most relevant behaviors. When only the recent interactions are leveraged, the long-term dependencies and periodicity might not be captured accurately. ETA-Net [164] aims to retrieve relevant items from long sequences in a more efficient manner. Instead of using a traditional dot-product operation, ETA-Net leverages a hashing-based target attention algorithm to calculate the Hamming distance, and then selects the top-k relevant behavior items for a target item. Finally, ETA-Net leverages multi-head attention to learn long sequence embeddings for capturing diverse interests from the selected top-k items. During prediction, ETA-Net leverages five different embeddings (i.e., long sequence embeddings, target item embeddings, short-sequence embeddings, user profile embeddings and context embeddings). For users with long interaction sequences, these sequences can be divided into sessions based on time intervals between interactions. As a result, HNUM [165] is designed with a two-layer architecture. The first layer is a session-level GRU model designed to capture users' short-term preferences within the current session. The second layer is a user-level memory network, which can capture users' long-term preferences across their entire interaction history. SIM [11] captures users' interests using both general search unit (GSU) and exact search unit (ESU). GSU leverages both hard search and soft search methods. The hard search aims to retrieve interacted items belonging to the same category of a candidate item from a user's long sequence. The soft search leverages inner product to identify top-k relevant items. These top-k items are used to construct users' sub-sequences. By employing multi-head attention, ESU effectively models the relationship between candidate items and the sub-sequences to extract users' long-term preferences. To solve the inconsistency between CP-GSU and ESU for target items, TWIN [166] leverages the same target-behavior relevance metric in both modules. Furthermore, to enhance efficiency, TWIN compresses user-item cross features into bias terms for target attention. To tackle ultra-long user behavior sequences, TWIN-V2 [167] leverages a hierarchical method to compress life-cycle behaviors. Similar items are classified into the same cluster. Meanwhile, TWIN-V2 leverages cluster-aware target attention to learn users' diverse interests. R-RNN [168] learns a user's global interests and recent interests based on global user behaviors and recent user behaviors, respectively. To improve efficiency while modeling long behavior sequences, SAM [169] leverages dual-query attention by regarding target item embedding and memory vector as queries. User interest center (UIC) is designed to decouple user interest modeling from the whole model. Modeling user interests needs to consume lots of computational time. UIC learns a user's latest interest representations according to real-time user behavior events. MIMN [170] leverages memory utilization regularization and memory induction unit to learn multi-channel user interests. UBCS [171] proposes a behavior sampling module to sample short sequences based on candidate items, relevance and temporal information. To speed up the process of sampling, UBCS leverages an item clustering module to cluster candidate items for selecting some representative candidate items. CoFARS [172] takes into account the influence of contexts on user preferences. CoFARS leverages a prototype-based approach to identify contexts that reflect similar user preferences. Then, CoFARS constructs a temporal graph containing context nodes and prototype nodes for integrating temporal information into users' interests. Finally, CoFARS identifies the prototypes that are consistent with the target context to generate a sub-sequence. Different from single-domain long sequential modeling, LCN [173] can transfer knowledge from a source domain to a target domain. In a cross representation production module, LCN leverages contrastive learning to bridge the connections among similar items in different domains. In a lifelong attention pyramid module, LCN leverages three kinds of cascading attentions to extract preferences from user lifelong behavior sequences considering candidate items. LRURec [174] leverages linear recurrent units and a recursive parallelization framework to enhance computational efficiency and realize low-cost inference. LinRec [175] reduces the quadratic time complexity of self-attention to a linear time complexity while preserving the advantages of attention mechanisms. To address the issue of sparse search behaviors failing to capture users' interests, QIN [176] adopts a two-stage search method. Firstly, the top-k items from a long sequence for user u are retrieved using a pre-trained retrieval model. Secondly, QIN predicts whether user u will purchase items based on the retrieved top-k items. For QIN, it first searches sub-sequences based on the relevance between behaviors and queries. Secondly, QIN searches sub-sequences according to the relevance between behaviors and target items. By considering item ID field and attribute field separately, QIN leverages a fused attention unit to learn users' representations better. To efficiently retrieve sub-sequences, SDIM [177] leverages an effective sampling method to generate sub-sequences. SDIM generates hash signatures for both the candidate item and each item in a user behavior sequence by multiple hash functions. SDIM can obtain the sub-sequences where items have the hash signatures the same as the candidate item. Item IDs might not exist in a dataset and different modal (i.e., text, images and attributes) embeddings are not aligned. Meanwhile, users' interacted item sequence and users' search query sequence reflect users' interests from different perspectives. To address these problems, SEMINAR [178] leverages a pre-training search unit to learn query-item pairs. Then, SEMINAR leverages pre-trained embeddings to initialize the network. Additionally, SEMINAR leverages a multi-modal product quantization strategy to reduce time complexity. The basic framework of SEMINAR includes GSU and ESU. To model users' long sequences and capture evolving users' interests, HPMN [179] leverages memory slots to realize sequential patterns' personalized memorization for each user. To capture multi-scale and evolving sequence patterns, HPMN leverages a hierarchical incremental updating mechanism.

6.5 Data-Augmented SR

There are two methods to conduct data augmentation. The first augmented methods only consider the information about item IDs. The second augmented methods that consider the role of side information. These approaches aim to alleviate the data sparsity problem and learn high-quality user representations. There are some SR models using data-augmented methods as follows.

Time intervals in users' interaction sequences might vary significantly, which might fail to model users' preferences due to preference drift. To solve this problem, TiCoSeRec [180] standardizes interaction sequences by transforming uneven time intervals into uniform distributions. TiCoSeRec consists of five operators (i.e., Ti-Crop, Ti-Reorder, Ti-Mask, Ti-Substitute and Ti-Insert). Different from traditional operators, the five data augmentation methods explicitly account for time intervals. Finally, TiCoSeRec leverages contrastive learning to ensure a high similarity between the augmented and original sequences. CL4SRec [12] leverages three kinds of data augmentation approaches (i.e., item crop, item mask and item reorder). Meanwhile, CL4SRec leverages contrastive learning between one augmented user interaction sequence and another augmented one. To alleviate the data sparsity problem and obtain stable users' interests, TGCL4SR [181] leverages two methods (i.e., neighbor sampling and time disturbance) to obtain augmented subgraphs. The neighbor sampling constructs augmented subgraphs by randomly sampling the neighbor nodes of each item. The time disturbance constructs the hypergraph to randomly add some noise that obeys Gaussian distribution to timestamps. Finally, by using disturbed graph contrastive loss and subgraph contrastive loss, high-quality representations can be obtained. Driven by the potential of data-centric AI, DR4SR+ [182] incorporates a model-aware dataset personalizer to tailor the regenerated dataset. In the process of model-agnostic dataset regeneration, a regenerator is pre-trained by learning transition patterns obtained through rule-based methods in the original interaction sequences. DR4SR+ leverages a diversity-promoted regenerator to capture one-to-many mapping relationships between the original sequences and generated patterns. To generate personalized datasets for target items, DR4SR+ uses a dataset personalizer to evaluate the quality of each generated sample for a target model. Traditional data augmentation methods often modify an original interaction sequence randomly. For example, some items are masked in sequences randomly. These methods might break sequential correlations. To solve this problem, MStein [183] leverages Wasserstein discrepancy measurement to measure mutual information among augmented sequences and ensure the maximum of the mutual information. MBASR [184] considers users' behaviors while data augmentation is conducted. DuoRec [185] leverages a contrastive learning regularizer to generate item embeddings with more uniform magnitude and frequency. END4Rec [186] improves the performance of Transformer by pruning noisy input. ASReP [187] leverages a pre-training Transformer to enhance the length of short users' interaction sequences, and then the Transformer is fine-tuned by augmented sequences.

7 Empirical Studies

In this section, our survey will introduce some datasets, evaluation protocols and evaluation metrics used in sequential recommendation.

7.1 Datasets

Training modality-based SR models with LLMs requires datasets that include multi-modal features. In our survey, we introduce some public datasets containing multi-modal features. Some specific details are shown in Table 4. These datasets primarily contain three different modals (i.e., text, images and videos). The strong sequential structure of these datasets is important for training SR models. Compared with other datasets in Table 4, Yelp and Steam might have weak sequential structure [188].

7.2 Evaluation Protocols

Datasets can be divided into training, validation and testing data using two primary methods. An early method is the leave-one-out strategy [1]. For this split method, all users' last interaction is split into the testing data, all users' penultimate interaction is split into the validation data, and the remaining interactions are split into the training data. This split method is simple and efficient, but it might lead to the problem of data leakage [189]. This is because interactions in the validation data may precede those in the training data based on their timestamps. The second method obtains the training, validation and testing data by specific timestamps [116]. We can set two timestamps (i.e., $T1$ and $T2$). All interactions happening before timestamp $T1$ are split into the training data. All interactions happening between timestamp $T1$ and timestamp $T2$ are split into the validation data. All interactions happening after timestamp $T2$ are assigned to the testing data. This splitting method helps to prevent data leakage. However, this splitting method has a shortcoming that some users might not be evaluated. Because all interacted records for some users could fall within training data.

During the validation and testing phase, candidate items can be selected by three methods (i.e., random sampling, sampling based on popularity, and full sampling). Random sampling mainly selects a small number of candidate items randomly [1]. This method is simple yet effective. Because the SR models are usually deployed in ranking or re-ranking phase, there are only a small number of

Table 4 The statistics about several widely used datasets.

Datasets	Multi-modal Features	Links
MIND	text	https://msnews.github.io
H-M	text, images	https://www.kaggle.com/c/h-and-m-personalized-fashion-recommendations/data
Bili	text, images	https://github.com/westlake-repl/NineRec?tab=readme-ov-file
Art of the Mix	text	https://brianmcfee.net/data/aotm2011.html
LastFM	text	https://www.kaggle.com/datasets/harshal19t/lastfm-dataset
Amazon 2023	text, images, videos	https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023
MicroLens	text, images, videos	https://github.com/westlake-repl/MicroLens
MovieLens	text	https://grouplens.org/datasets/movielens/
Steam	text, images	https://huggingface.co/datasets/FronkonGames/steam-games-dataset
Yelp	text, images	https://www.yelp.com/dataset
Goodreads	text, images	https://www.kaggle.com/datasets/jealousleopard/goodreadsbooks
Douban	text	https://www.kaggle.com/datasets/fengzhujoey/douban-datasetratingreviewside-information

candidate items. The number of candidate items is consistent with the actual situation. However, random sampling might lead to variability in the candidate items. The evaluation results might be unstable with the varying candidate items. Sampling based on popularity is that the candidate items are selected based on popularity [103]. If the popularity of items is larger, the probability of these items that are selected as candidate items is larger. This method can reduce the influence of randomness. Users interact with popular items easily, therefore these items appear in candidate items more frequently. Compared with random sampling, this split method is more stable and provides a more accurate evaluation of recommendation performance. The final method considers all items in the dataset as candidate items [182]. This method can evaluate the recommendation performance of different models from a comprehensive view. However, it might consume lots of computational resources especially when the dataset contains a large number of items.

7.3 Evaluation Metrics

SR models aim to generate a top-N recommendation list. The evaluation metrics mainly aim to measure whether the user's most recently interacted item is in the top-N recommendation list and the position in the top-N recommendation list. Two commonly used evaluation metrics [1,18] are shown as follows.

Hit Ratio (HR) denotes the ratio of ground-truth items included in the top-N recommendation list. It can be formalized as:

$$HR@N = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} \sigma(R_{u,t_u} \leq N) \tag{26}$$

where $\sigma(\cdot)$ is an indicator function. If the rank generated by an SR model is among top-N, the value of $\sigma(\cdot)$ is 1, otherwise is 0. Normalized Discounted Cumulative Gain (NDCG) evaluates the quality of the ranking by considering both the relevance and position of ground-truth items in the top-N recommendation list. It can be formalized as:

$$NDCG@N = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\sigma(R_{u,t_u} \leq N)}{\log_2(1 + R_{u,t_u})} \tag{27}$$

8 Prospects and Future Directions

The SR models have been studied for several years, but there are certain areas that remain to be fully explored.

8.1 Open-Domain SR

Traditional SR models are trained on a single and static dataset with predefined features. When new items emerge frequently and data resources are increasingly diverse, the traditional SR models are difficult to recommend personalized items. Leveraging the multi-modal features of items and LLMs, open-domain SR models can operate effectively in dynamic environments with a continuous influx of new items. Meanwhile, the open-domain SR models can be transferred across different platforms and domains. By transferring knowledge from external domains or platforms, the open-domain SR models can achieve better recommendation performance within the target domain or platform.

8.2 Data-Centric SR

To improve the recommendation performance, data-centric SR models mainly focus on improving the quality of data rather than refining model structures. The data is important in sequential recommendation tasks. The data-centric SR models mainly leverage three techniques (i.e., data generation, data denoising and data debiasing). The data generation has been researched for many years. However, challenges remain in generating high-quality data for certain existing datasets. The quality of the generated data cannot be measured by a specific method. Some SR models leverage data augmentation to generate additional data, aiming to improve the recommendation performance. However, SR models need to be trained in a larger number of data, which might consume more computational resources and training time. Therefore, we need to design a data augmentation method to improve the quality of generated data and reduce the training time. In the meantime, we can consider some multi-modal features to generate some high-quality data.

8.3 Cloud-Edge Collaborative SR

Cloud-edge collaborative SR models combine the strengths of cloud-based and edge-based SR models. Compared with conventional SR models, cloud-edge collaborative SR models can not only improve the recommendation performance but also protect user privacy and leverage resources more effectively. The conventional SR models often require lots of computational resources. Therefore, the conventional SR models are hard to deploy on their own computers. Moreover, transmitting interaction data to a server could risk exposing private information. Therefore, the cloud-edge collaborative SR models are meaningful. Federated SR models are one such approach specifically designed to protect privacy.

8.4 Continuous SR

In many short-video scenarios, such as TikTok, the length of a user's interaction sequence tends to grow over time. Some ultra-long interaction sequences might influence the efficiency of SR models. Therefore, how to design an SR model that can handle ultra-long interaction sequences is a challenge. Meanwhile, a user's interests might evolve over time. Therefore, continuous SR models need to update the user's interests dynamically based on recent interactions.

8.5 SR for Good

Some existing SR models aim to improve economic growth. However, SR models for good aim to recommend personalized items benefit both individuals and society while also supporting economic development. For example, to reduce carbon emissions, it is important to recommend environmentally friendly items for users. In educational field, SR models should recommend learning materials to help users learn more effectively. In healthy field, SR models should recommend personalized exercises to help users improve their physical fitness and overall health. The SR models for good remain to be explored.

8.6 Explainable SR

Some SR models based on multi-modal features and large language models focus on improving the recommendation accuracy but often fail to provide explanations for why these related items are recommended. LLMs excel at processing and understanding multi-modal features (i.e., text, images and so on), which provides an opportunity to generate some explanations. Therefore, in future work, we can leverage large language models and multi-modal features of interacted items to offer more interpretation in recommendations.

9 Conclusions

In this survey, we have shown that SR models have achieved significant advancements in recommending personalized items for users. From their development, we can have some observations. Firstly, modal-based SR models can tackle the data sparsity and cold-start problems better than pure ID-based SR models. Secondly, modal-based SR models are of high time complexity and need to consume more training time than pure ID-based SR models. Thirdly, modal-based SR models are often associated with better transferability than pure ID-based SR models and SR models combining IDs and general features. Fourthly, multi-modal SR models using IDs can learn fine-grained preferences and achieve more accurate recommendation performance. Lastly, incorporating recent techniques like LLMs into SR models offers further potential for improved recommendation effectiveness.

Acknowledgements We thank the support of National Natural Science Foundation of China (Nos. 62461160311, 62172283 and 62272315), and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515010122).

References

1. Kang W C, McAuley J. Self-attentive sequential recommendation. In: Proceedings of the 18th IEEE International Conference on Data Mining. 2018, 197–206
2. He R, Kang W C, McAuley J. Translation-based recommendation. In: Proceedings of the 11th ACM Conference on Recommender Systems. 2017, 161–169
3. Yuan X, Duan D, Tong L, Shi L, Zhang C. ICAT-SR: Item categorical attribute integrated sequential recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, 1687–1691
4. Liu C, Li X, Cai G, Dong Z, Zhu H, Shang L. Non-invasive self-attention for side information fusion in sequential recommendation. In: Proceedings of the AAAI conference on artificial intelligence. 2021, 4249–4256
5. Kenton J D M W C, Toutanova L K. BERT: Pre-training of deep bidirectional Transformers for language understanding. In: Proceedings of Proceedings of the 17-th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019, 4171–4186
6. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, 770–778

7. Yuan Z, Yuan F, Song Y, Li Y, Fu J, Yang F, Pan Y, Ni Y. Where to go next for recommender systems? ID- vs. modality-based recommender models revisited. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023, 2639–2649
8. Hou Y, Mu S, Zhao W X, Li Y, Ding B, Wen J R. Towards universal sequence representation learning for recommender systems. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022, 585–593
9. Ji J, Li Z, Xu S, Hua W, Ge Y, Tan J, Zhang Y. GenRec: Large language model for generative recommendation. In: Proceedings of the 46th European Conference on Information Retrieval. 2024, 494–502
10. Boz A, Zorgdrager W, Kotti Z, Harte J, Louridas P, Jannach D, Fragkoulis M. Improving sequential recommendations with LLMs. arXiv preprint arXiv:2402.01339, 2024
11. Pi Q, Zhou G, Zhang Y, Wang Z, Ren L, Fan Y, Zhu X, Gai K. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. 2020, 2685–2692
12. Xie X, Sun F, Liu Z, Wu S, Gao J, Zhang J, Ding B, Cui B. Contrastive learning for sequential recommendation. In: Proceedings of the 38th International Conference on Data Engineering. 2022, 1259–1273
13. Wang S, Cao L, Wang Y, Sheng Q Z, Orgun M A, Lian D. A survey on session-based recommender systems. ACM Computing Surveys, 2021, 54(7): 1–38
14. Li Z, Yang C, Chen Y, Wang X, Chen H, Xu G, Yao L, Sheng M. Graph and sequential neural networks in session-based recommendation: A survey. ACM Computing Surveys, 2024, 57(2): 1–37
15. Quadrana M, Cremonesi P, Jannach D. Sequence-aware recommender systems. ACM Computing Surveys, 2018, 51(4): 1–36
16. Nasir M, Ezeife C I. A survey and taxonomy of sequential recommender systems for e-commerce product recommendation. SN Computer Science, 2023, 4(6): 708
17. Fang H, Zhang D, Shu Y, Guo G. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. ACM Transactions on Information Systems, 2020, 39(1): 1–42
18. Boka T F, Niu Z, Neupane R B. A survey of sequential recommendation systems: Techniques, evaluation, and future directions. Information Systems, 2024, 102427
19. Chen X, Li Z, Pan W, Ming Z. A survey on multi-behavior sequential recommendation. arXiv preprint arXiv:2308.15701, 2023
20. Chen S, Xu Z, Pan W, Yang Q, Ming Z. A survey on cross-domain sequential recommendation. In: Proceedings of the 33th International Joint Conference on Artificial Intelligence. 2024, 7989–7998
21. Jing M, Zhu Y, Zang T, Wang K. Contrastive self-supervised learning in recommender systems: A survey. ACM Transactions on Information Systems, 2023, 42(2): 1–39
22. Lai R, Chen L, Chen R, Zhang C. A survey on data-centric recommender systems. arXiv preprint arXiv:2401.17878, 2024
23. Zhang X, Xu B, Li C, Zhou Y, Li L, Lin H. Side information-driven session-based recommendation: A survey. arXiv preprint arXiv:2402.17129, 2024
24. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D. Session-based recommendations with recurrent neural networks. In: Proceedings of the 4th International Conference on Learning Representations. 2015
25. Xie Y, Zhou P, Kim S. Decoupled side information fusion for sequential recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 1611–1621
26. Li Y, Ding Y, Chen B, Xin X, Wang Y, Shi Y, Tang R, Wang D. Extracting attentive social temporal excitation for sequential recommendation. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management. 2021, 998–1007
27. Hou Y, He Z, McAuley J, Zhao W X. Learning vector-quantized item representation for transferable sequential recommenders. In: Proceedings of the ACM Web Conference 2023. 2023, 1162–1171
28. Mobasher B, Dai H, Luo T, Nakagawa M. Using sequential and non-sequential patterns in predictive web usage mining tasks. In: Proceedings of the 2th IEEE International Conference on Data Mining. 2002, 669–672
29. Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the ACM Web Conference 2010. 2010, 811–820
30. Hidasi B, Karatzoglou A. Recurrent neural networks with top-k gains for session-based recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018, 843–852
31. Lin X, Niu S, Wang Y, Li Y. K-plet recurrent neural networks for sequential recommendation. In: Proceedings of the 41st International ACM SIGIR Conference

on Research and Development in Information Retrieval. 2018, 1057–1060

32. Niu S, Zhang R. Collaborative sequence prediction for sequential recommender. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 2017, 2239–2242

33. Chen Q, Li G, Zhou Q, Shi S, Zou D. Double attention convolutional neural network for sequential recommendation. ACM Transactions on the Web, 2022, 16(4): 1–23

34. 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

35. Yan A, Cheng S, Kang W C, Wan M, McAuley J. CosRec: 2D convolutional neural networks for sequential recommendation. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019, 2173–2176

36. Sun F, Liu J, Wu J, Pei C, Lin X, Ou W, Jiang P. 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

37. Petrov A, Macdonald C. A systematic review and replicability study of bert4rec for sequential recommendation. In: Proceedings of the 16th ACM Conference on Recommender Systems. 2022, 436–447

38. Zhang Y, Wang X, Chen H, Zhu W. Adaptive disentangled Transformer for sequential recommendation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023, 3434–3445

39. Zhou P, Ye Q, Xie Y, Gao J, Wang S, Kim J B, You C, Kim S. Attention calibration for Transformer-based sequential recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023, 3595–3605

40. Liu Z, Liu S, Zhang Z, Cai Q, Zhao X, Zhao K, Hu L, Jiang P, Gai K. Sequential recommendation for optimizing both immediate feedback and long-term retention. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024, 1872–1882

41. Fan Z, Liu Z, Wang Y, Wang A, Nazari Z, Zheng L, Peng H, Yu P S. Sequential recommendation via stochastic self-attention. In: Proceedings of the ACM Web Conference 2022. 2022, 2036–2047

42. Ren R, Liu Z, Li Y, Zhao W X, Wang H, Ding B, Wen J R. Sequential recommendation with self-attentive multi-adversarial network. In: Proceedings of the 43rd International ACM SIGIR conference on Research and Development in Information Retrieval. 2020, 89–98

43. Wu L, Li S, Hsieh C J, Sharpnack J. SSE-PT: Sequential recommendation via personalized Transformer. In: Proceedings of the 14th ACM conference on recommender systems. 2020, 328–337

44. Li C, Wang Y, Liu Q, Zhao X, Wang W, Wang Y, Zou L, Fan W, Li Q. STRec: Sparse Transformer for sequential recommendations. In: Proceedings of the 17th ACM Conference on Recommender Systems. 2023, 101–111

45. Cui Z, Cai Y, Wu S, Ma X, Wang L. Motif-aware sequential recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, 1738–1742

46. Guo N, Liu X, Li S, Ma Q, Gao K, Han B, Zheng L, Guo S, Guo X. Poincaré heterogeneous graph neural networks for sequential recommendation. ACM Transactions on Information Systems, 2023, 41(3): 1–26

47. Du H, Shi H, Zhao P, Wang D, Sheng V S, Liu Y, Liu G, Zhao L. Contrastive learning with bidirectional Transformers for sequential recommendation. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management. 2022, 396–405

48. Yang Y, Huang C, Xia L, Huang C, Luo D, Lin K. Debiased contrastive learning for sequential recommendation. In: Proceedings of the ACM Web Conference 2023. 2023

49. Lin G, Gao C, Zheng Y, Chang J, Niu Y, Song Y, Li Z, Jin D, Li Y. Dual-interest factorization-heads attention for sequential recommendation. In: Proceedings of the ACM Web Conference 2023. 2023, 917–927

50. Xu C, Xu J, Chen X, Dong Z, Wen J R. Dually enhanced propensity score estimation in sequential recommendation. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management. 2022, 2260–2269

51. Ma M, Ren P, Chen Z, Ren Z, Liang H, Ma J, De Rijke M. Improving Transformer-based sequential recommenders through preference editing. ACM Transactions on Information Systems, 2023, 41(3): 1–24

52. Xi X, Zhao Y, Liu Q, Ouyang L, Wu Y. Integrating offline reinforcement learning with Transformers for sequential recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems. 2023, 1–1

53. Fan Z, Liu Z, Wang S, Zheng L, Yu P S. Modeling sequences as distributions with uncertainty for sequential recommendation. In: Proceedings of the 30th ACM In-

ternational Conference on Information and Knowledge Management. 2021, 3019–3023

54. Souza Pereira Moreira d G, Rabhi S, Lee J M, Ak R, Oldridge E. Transformers4Rec: Bridging the gap between nlp and sequential/session-based recommendation. In: Proceedings of the 15th ACM Conference on Recommender Systems. 2021, 143–153
55. Zhu Y, Huang B, Jiang S, Yang M, Yang Y, Zhong W. Progressive self-attention network with unsymmetrical positional encoding for sequential recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 2029–2033
56. Lai V, Chen H, Yeh C C M, Xu M, Cai Y, Yang H. Enhancing Transformers without self-supervised learning: A loss landscape perspective in sequential recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems. 2023, 791–797
57. Liu Q, Yan F, Zhao X, Du Z, Guo H, Tang R, Tian F. Diffusion augmentation for sequential recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023, 1576–1586
58. Li Z, Sun A, Li C. DiffuRec: A diffusion model for sequential recommendation. ACM Transactions on Information Systems, 2023, 42(3): 1–28
59. Wu Z, Wang X, Chen H, Li K, Han Y, Sun L, Zhu W. Diff4Rec: Sequential recommendation with curriculum-scheduled diffusion augmentation. In: Proceedings of the 31st ACM International Conference on Multimedia. 2023, 9329–9335
60. Liu S, Liu J, Gu H, Li D, Lu T, Zhang P, Gu N. AutoSeqRec: Autoencoder for efficient sequential recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023, 1493–1502
61. Chen X, Li Q. Causality-driven user modeling for sequential recommendations over time. In: Companion Proceedings of the ACM on Web Conference 2024. 2024, 1400–1406
62. Ding Y, Ma Y, Wong W K, Chua T S. Leveraging two types of global graph for sequential fashion recommendation. In: Proceedings of the 2021 International Conference on Multimedia Retrieval. 2021, 73–81
63. Peng B, Chen Z, Parthasarathy S, Ning X. Modeling sequences as star graphs to address over-smoothing in self-attentive sequential recommendation. ACM Transactions on Knowledge Discovery from Data, 2023
64. Zhou K, Yu H, Zhao W X, Wen J R. Filter-enhanced MLP is all you need for sequential recommendation. In: Proceedings of the ACM Web Conference 2022. 2022, 2388–2399
65. Jiang Y, Xu Y, Yang Y, Yang F, Wang P, Li C, Zhuang F, Xiong H. TriMLP: A foundational MLP-like architecture for sequential recommendation. ACM Transactions on Information Systems, 2024, 42(6)
66. Long C, Yuan H, Fang J, Xian X, Liu G, Sheng V S, Zhao P. Learning global and multi-granularity local representation with MLP for sequential recommendation. ACM Transactions on Knowledge Discovery from Data, 2024, 18(4): 1–15
67. Sachdeva N, Manco G, Ritacco E, Pudi V. Sequential variational autoencoders for collaborative filtering. In: Proceedings of the 12th ACM International Conference on Web Search and Data Mining. 2019, 600–608
68. You J, Wang Y, Pal A, Eksombatchai P, Rosenburg C, Leskovec J. Hierarchical temporal convolutional networks for dynamic recommender systems. In: Proceedings of the ACM Web Conference 2019. 2019, 2236–2246
69. Zhang Q, Wu B, Sun Z, Ye Y. Gating augmented capsule network for sequential recommendation. Knowledge-Based Systems, 2022, 247: 108817
70. Wang Z, Shen Y. Time-aware multi-interest capsule network for sequential recommendation. In: Proceedings of the 2022 SIAM International Conference on Data Mining. 2022, 558–566
71. Zhao X, Zhang L, Xia L, Ding Z, Yin D, Tang J. Deep reinforcement learning for list-wise recommendations. arXiv preprint arXiv:1801.00209, 2017
72. Antaris S, Rafailidis D. Sequence adaptation via reinforcement learning in recommender systems. In: Proceedings of the 15th ACM Conference on Recommender Systems. 2021, 714–718
73. Jiang J, Zhang P, Luo Y, Li C, Kim J B, Zhang K, Wang S, Xie X, Kim S. AdaMCT: Adaptive mixture of CNN-Transformer for sequential recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023, 976–986
74. Wang J, Liu Q, Liu Z, Wu S. Towards accurate and interpretable sequential prediction: A CNN & attention-based feature extractor. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019, 1703–1712
75. Huang X, Sang J, Yu J, Xu C. Learning to learn a cold-start sequential recommender. ACM Transactions on Information Systems, 2022, 40(2): 1–25
76. Chen H, Huang B, Wang X, Zhou Y, Zhu W. Global-local graphformer: Towards better understanding of user intentions in sequential recommendation. In: Proceedings of the 5th ACM International Conference on

Multimedia in Asia. 2023, 1–7

77. Ye Y, Xia L, Huang C. Graph masked autoencoder for sequential recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023, 321–330

78. Zhang W, Chen Z, Zha H, Wang J. Learning from substitutable and complementary relations for graph-based sequential product recommendation. ACM Transactions on Information Systems, 2021, 40(2): 1–28

79. Yin M, Wang H, Xu X, Wu L, Zhao S, Guo W, Liu Y, Tang R, Lian D, Chen E. APGL4SR: A generic framework with adaptive and personalized global collaborative information in sequential recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023, 3009–3019

80. Baikalov V, Frolov E. End-to-end graph-sequential representation learning for accurate recommendations. In: Companion Proceedings of the ACM on Web Conference 2024. 2024, 501–504

81. Tan L, Xu J, Gong D, Liu F. Recurrent attentive neural networks for sequential recommendation. In: Proceedings of the 2023 International Conference on Communication Network and Machine Learning. 2023, 356–360

82. Xu C, Zhao P, Liu Y, Xu J, S. Sheng V S S, Cui Z, Zhou X, Xiong H. Recurrent convolutional neural network for sequential recommendation. In: Proceedings of the ACM Web Conference 2019. 2019, 3398–3404

83. Yu M, Zhu K, Zhao M, Yu J, Xu T, Jin D, Li X, Yu R. Learning neighbor user intention on user–item interaction graphs for better sequential recommendation. ACM Transactions on the Web, 2024, 18(2): 1–28

84. Tang J, Belletti F, Jain S, Chen M, Beutel A, Xu C, H. Chi E. Towards neural mixture recommender for long range dependent user sequences. In: Proceedings of the ACM Web Conference 2019. 2019, 1782–1793

85. Alharbi N, Caragea D. Cross-domain self-attentive sequential recommendations. In: Proceedings of the 3rd International Conference on Data Science and Applications. 2022, 601–614

86. Ma H, Xie R, Meng L, Chen X, Zhang X, Lin L, Zhou J. Triple sequence learning for cross-domain recommendation. ACM Transactions on Information Systems, 2024, 42(4): 1–29

87. Cao J, Cong X, Sheng J, Liu T, Wang B. Contrastive cross-domain sequential recommendation. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management. 2022, 138–147

88. Li J, Wang Y, McAuley J. Time interval aware self-attention for sequential recommendation. In: Proceedings of the 13th ACM International Conference on Web Search and Data Mining. 2020, 322–330

89. Cho J, Hyun D, Kang S, Yu H. Learning heterogeneous temporal patterns of user preference for timely recommendation. In: Proceedings of the ACM Web Conference 2021. 2021, 1274–1283

90. Yuan E, Guo W, He Z, Guo H, Liu C, Tang R. Multi-behavior sequential Transformer recommender. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 1642–1652

91. Su J, Chen C, Lin Z, Li X, Liu W, Zheng X. Personalized behavior-aware Transformer for multi-behavior sequential recommendation. In: Proceedings of the 31st ACM International Conference on Multimedia. 2023, 6321–6331

92. Li X, Chen H, Yu J, Zhao M, Xu T, Zhang W, Yu M. 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

93. Chen Z, Zhang W, Yan J, Wang G, Wang J. Learning dual dynamic representations on time-sliced user-item interaction graphs for sequential recommendation. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management. 2021, 231–240

94. Tran V A, Salha-Galvan G, Sguerra B, Hennequin R. Attention mixtures for time-aware sequential recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023, 1821–1826

95. Fan Z, Liu Z, Zhang J, Xiong Y, Zheng L, Yu P S. Continuous-time sequential recommendation with temporal graph collaborative Transformer. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management. 2021, 433–442

96. Chen Z, Wang W. Time interval-aware graph with self-attention for sequential recommendation. In: Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence. 2023, 1–9

97. Du X, Li X. Moving forward together: A multi-user pattern extraction neural model for sequential recommendations. In: Proceedings of the 6th International Conference on Artificial Intelligence and Pattern Recognition. 2023, 1355–1361

98. Wang J, Ding K, Hong L, Liu H, Caverlee J. Next-item recommendation with sequential hypergraphs. In:

Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020, 1101–1110

99. Huang L, Ma Y, Liu Y, Danny Du B, Wang S, Li D. Position-enhanced and time-aware graph convolutional network for sequential recommendations. ACM Transactions on Information Systems, 2023, 41(1): 1–32

100. Zhang Y, He Y, Wang J, Caverlee J. Adaptive hierarchical translation-based sequential recommendation. In: Proceedings of the Web Conference 2020. 2020, 2984–2990

101. Huang C, Wang S, Wang X, Yao L. Dual contrastive Transformer for hierarchical preference modeling in sequential recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023, 99–109

102. Li M, Zhang Z, Zhao X, Wang W, Zhao M, Wu R, Guo R. AutoMLP: Automated MLP for sequential recommendations. In: Proceedings of the ACM Web Conference 2023. 2023, 1190–1198

103. Xu Z, Pan W, Ming Z. A multi-view graph contrastive learning framework for cross-domain sequential recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems. 2023, 491–501

104. Zheng X, Su J, Liu W, Chen C. DDGHM: Dual dynamic graph with hybrid metric training for cross-domain sequential recommendation. In: Proceedings of the 30th ACM International Conference on Multimedia. 2022, 471–481

105. Li Y, Li Q, Meng S, Hou J. Transformer-based rating-aware sequential recommendation. In: Proceedings of the 21th International Conference on Algorithms and Architectures for Parallel Processing. 2021, 759–774

106. Zhang X, Xu B, Yang L, Li C, Ma F, Liu H, Lin H. Price does matter! modeling price and interest preferences in session-based recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 1684–1693

107. Huang X, Fang Q, Qian S, Sang J, Li Y, Xu C. Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. In: Proceedings of the 27th ACM International Conference on Multimedia. 2019, 548–556

108. Huang J, Zhao W X, Dou H, Wen J R, Chang E Y. Improving sequential recommendation with knowledge-enhanced memory networks. In: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. 2018, 505–514

109. Zhai J, Zheng X, Wang C D, Li H, Tian Y. Knowledge prompt-tuning for sequential recommendation. In: Proceedings of the 31st ACM International Conference on Multimedia. 2023, 6451–6461

110. Wang P, Fan Y, Xia L, Zhao W X, Niu S, Huang J. KERL: A knowledge-guided reinforcement learning model for sequential recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020, 209–218

111. Hu H, Guo W, Liu X, Liu Y, Tang R, Zhang R, Kan M Y. User behavior enriched temporal knowledge graphs for sequential recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining. 2024, 266–275

112. Chen J, Cao Y, Zhang F, Sun P, Wei K. Sequential intention-aware recommender based on user interaction graph. In: Proceedings of the 2022 International Conference on Multimedia Retrieval. 2022, 118–126

113. Chen T, Wong R C W. An efficient and effective framework for session-based social recommendation. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. 2021, 400–408

114. Li C, Niu X, Luo X, Chen Z, Quan C. A review-driven neural model for sequential recommendation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019, 2866–2872

115. Zhang J, Mu X, Zhao P, Kang K, Ma C. Improving current interest with item and review sequential patterns for sequential recommendation. Engineering Applications of Artificial Intelligence, 2021, 104: 104348

116. Huang B, Luo J, Du W, Pan W, Ming Z. Cascaded cross attention for review-based sequential recommendation. In: Proceedings of the 23th IEEE International Conference on Data Mining. 2023, 170–179

117. Li M, Zhao X, Lyu C, Zhao M, Wu R, Guo R. MLP4Rec: A pure MLP architecture for sequential recommendations. arXiv preprint arXiv:2204.11510, 2022

118. Gao J, Zhao X, Li M, Zhao M, Wu R, Guo R, Liu Y, Yin D. SMLP4Rec: An efficient all-MLP architecture for sequential recommendations. ACM Transactions on Information Systems, 2024, 42(3): 1–23

119. Hsu C, Li C T. RetaGNN: Relational temporal attentive graph neural networks for holistic sequential recommendation. In: Proceedings of The Web Conference 2021. 2021, 2968–2979

120. Zhang T, Zhao P, Liu Y, Sheng V S, Xu J, Wang D, Liu G, Zhou X. Feature-level deeper self-attention network for sequential recommendation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019, 4320–4326

121. Hu H, Guo W, Liu Y, Kan M Y. Adaptive multi-

modalities fusion in sequential recommendation systems. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023, 843–853

122. Wang S, Shen B, Min X, He Y, Zhang X, Zhang L, Zhou J, Mo L. Aligned side information fusion method for sequential recommendation. In: Companion Proceedings of the ACM on Web Conference 2024. 2024, 112–120

123. Zhou X, Lumbantoruan R, Ren Y, Chen L, Yang X, Shao J. Dynamic bi-layer graph learning for context-aware sequential recommendation. ACM Transactions on Recommender Systems, 2024, 2(2): 1–23

124. Smirnova E, Vasile F. Contextual sequence modeling for recommendation with recurrent neural networks. In: Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems. 2017, 2–9

125. Rakkappan L, Rajan V. Context-aware sequential recommendations withstacked recurrent neural networks. In: Proceedings of the ACM Web Conference 2019. 2019, 3172–3178

126. Zhang J, Lin F, Yang C, Jiang W. A new sequential prediction framework with spatial-temporal embedding. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 2282–2286

127. Xue L, Yang D, Zhai S, Li Y, Xiao Y. Learning dual-view user representations for enhanced sequential recommendation. ACM Transactions on Information Systems, 2023, 41(4): 1–26

128. Zheng L, Chai H, Chen X, Jin J, Zhang W, Yu Y, Guo X, Ge C, Feng Z. Search-based time-aware graph-enhanced recommendation with sequential behavior data. ACM Transactions on Recommender Systems, 2024, 2(4): 1–29

129. Li H, Yu L, Niu X, Leng Y, Du Q. Sequential and graphical cross-domain recommendations with a multi-view hierarchical transfer gate. ACM Transactions on Knowledge Discovery from Data, 2023, 18(1): 1–28

130. Ji W, Liu X, Zhang A, Wei Y, Ni Y, Wang X. Online distillation-enhanced multi-modal Transformer for sequential recommendation. In: Proceedings of the 31st ACM International Conference on Multimedia. 2023, 955–965

131. Zhang J, Cheng Y, Ni Y, Pan Y, Yuan Z, Fu J, Li Y, Wang J, Yuan F. Ninerec: A benchmark dataset suite for evaluating transferable recommendation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024

132. Mu S, Hou Y, Zhao W X, Li Y, Ding B. Id-agnostic user behavior pre-training for sequential recommendation. In: Proceedings of the 28th China Conference on Information Retrieval. 2022, 16–27

133. Ni Y, Cheng Y, Liu X, Fu J, Li Y, He X, Zhang Y, Yuan F. A content-driven micro-video recommendation dataset at scale. arXiv preprint arXiv:2309.15379, 2023

134. Qu Z, Xie R, Xiao C, Sun X, Kang Z. The elephant in the room: Rethinking the usage of pre-trained language model in sequential recommendation. arXiv preprint arXiv:2404.08796, 2024

135. Zivic P, Vazquez H, Sánchez J. Scaling sequential recommendation models with Transformers. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024, 1567–1577

136. Li J, Wang M, Li J, Fu J, Shen X, Shang J, McAuley J. Text is all you need: Learning language representations for sequential recommendation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023, 1258–1267

137. Liang J, Zhao X, Li M, Zhang Z, Wang W, Liu H, Liu Z. Mmmlp: Multi-modal multilayer perceptron for sequential recommendations. In: Proceedings of the ACM Web Conference 2023. 2023, 1109–1117

138. Hidasi B, Quadrana M, Karatzoglou A, Tikk D. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems. 2016, 241–248

139. Rajput S, Mehta N, Singh A, Hulikal Keshavan R, Vu T, Heldt L, Hong L, Tay Y, Tran V, Samost J, others. Recommender systems with generative retrieval. Advances in Neural Information Processing Systems, 2024, 36

140. Liu Z, Hou Y, McAuley J. Multi-behavior generative recommendation. arXiv preprint arXiv:2405.16871, 2024

141. Wang W, Bao H, Lin X, Zhang J, Li Y, Feng F, Ng S K, Chua T S. Learnable item tokenization for generative recommendation. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2024, 2400–2409

142. Wang Y, Xun J, Hong M, Zhu J, Jin T, Lin W, Li H, Li L, Xia Y, Zhao Z, others . EAGER: Two-stream generative recommender with behavior-semantic collaboration. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024, 3245–3254

143. Zhu J, Jin M, Liu Q, Qiu Z, Dong Z, Li X. CoST: Contrastive quantization based semantic tokenization for generative recommendation. In: Proceedings of the 18th ACM Conference on Recommender Systems. 2024, 969–974

144. Liu H, Wei Y, Song X, Guan W, Li Y F, Nie L. MMGRec: Multimodal generative recommendation with Transformer model. arXiv preprint arXiv:2404.16555, 2024

145. Geng S, Liu S, Fu Z, Ge Y, Zhang Y. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (P5). In: Proceedings of the 16th ACM Conference on Recommender Systems. 2022, 299–315

146. Wang H, Liu X, Fan W, Zhao X, Kini V, Yadav D, Wang F, Wen Z, Tang J, Liu H. Rethinking large language model architectures for sequential recommendations. arXiv preprint arXiv:2402.09543, 2024

147. Wang Y, Tian C, Hu B, Yu Y, Liu Z, Zhang Z, Zhou J, Pang L, Wang X. Can small language models be good reasoners for sequential recommendation? In: Proceedings of the ACM on Web Conference 2024. 2024, 3876–3887

148. Zheng Z, Chao W, Qiu Z, Zhu H, Xiong H. Harnessing large language models for text-rich sequential recommendation. In: Proceedings of the ACM on Web Conference 2024. 2024, 3207–3216

149. Yang Z, Wu J, Luo Y, Zhang J, Yuan Y, Zhang A, Wang X, He X. Large language model can interpret latent space of sequential recommender. arXiv preprint arXiv:2310.20487, 2023

150. Bao K, Zhang J, Zhang Y, Wang W, Feng F, He X. TALLRec: An effective and efficient tuning framework to align large language model with recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems. 2023, 1007–1014

151. Cui Y, Liu F, Wang P, Wang B, Tang H, Wan Y, Wang J, Chen J. Distillation matters: Empowering sequential recommenders to match the performance of large language model. arXiv preprint arXiv:2405.00338, 2024

152. Li X, Chen C, Zhao X, Zhang Y, Xing C. E4SRec: An elegant effective efficient extensible solution of large language models for sequential recommendation. arXiv preprint arXiv:2312.02443, 2023

153. Shen T, Wang H, Zhang J, Zhao S, Li L, Chen Z, Lian D, Chen E. Exploring user retrieval integration towards large language models for cross-domain sequential recommendation. arXiv preprint arXiv:2406.03085, 2024

154. Ren Y, Chen Z, Yang X, Li L, Jiang C, Cheng L, Zhang B, Mo L, Zhou J. Enhancing sequential recommenders with augmented knowledge from aligned large language models. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024, 345–354

155. Hu J, Xia W, Zhang X, Fu C, Wu W, Huan Z, Li A, Tang Z, Zhou J. Enhancing sequential recommendation via LLM-based semantic embedding learning. In: Companion Proceedings of the ACM on Web Conference 2024. 2024, 103–111

156. Zheng B, Lin Z, Liu E, Yang C, Bai E, Ling C, Zhao W X, Wen J R. A large language model enhanced sequential recommender for joint video and comment recommendation. arXiv preprint arXiv:2403.13574, 2024

157. Liu Q, Wu X, Zhao X, Wang Y, Zhang Z, Tian F, Zheng Y. Large language models enhanced sequential recommendation for long-tail user and item. arXiv preprint arXiv:2405.20646, 2024

158. Zhang X, Xu B, Wu Y, Zhong Y, Lin H, Ma F. FineRec: Exploring fine-grained sequential recommendation. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024, 1599–1608

159. Yang S, Ma W, Sun P, Ai Q, Liu Y, Cai M, Zhang M. Sequential recommendation with latent relations based on large language model. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024, 335–344

160. Lin J, Shan R, Zhu C, Du K, Chen B, Quan S, Tang R, Yu Y, Zhang W. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In: Proceedings of the ACM on Web Conference 2024. 2024, 3497–3508

161. Zhu J, Lin J, Dai X, Chen B, Shan R, Zhu J, Tang R, Yu Y, Zhang W. Lifelong personalized low-rank adaptation of large language models for recommendation. arXiv preprint arXiv:2408.03533, 2024

162. Geng B, Huan Z, Zhang X, He Y, Zhang L, Yuan F, Zhou J, Mo L. Breaking the length barrier: LLM-enhanced CTR prediction in long textual user behaviors. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024, 2311–2315

163. Qin J, Zhang W, Wu X, Jin J, Fang Y, Yu Y. User behavior retrieval for click-through rate prediction. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020, 2347–2356

164. Chen Q, Xu Y, Pei C, Lv S, Zhuang T, Ge J. Efficient long sequential user data modeling for click-through rate prediction. arXiv preprint arXiv:2209.12212, 2022

165. Dong J, Sun F, Wu T, Wu X, Zhang W, Wang S. A hierarchical network with user memory matrix for long sequence recommendation. Wireless Communications and Mobile Computing, 2022, 2022(1): 5457044

166. Chang J, Zhang C, Fu Z, Zang X, Guan L, Lu J, Hui Y, Leng D, Niu Y, Song Y, others. TWIN: Two-stage

interest network for lifelong user behavior modeling in CTR prediction at KuaiShou. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023, 3785–3794

167. Si Z, Guan L, Sun Z, Zang X, Lu J, Hui Y, Cao X, Yang Z, Zheng Y, Leng D, others. TWIN V2: Scaling ultra-long user behavior sequence modeling for enhanced CTR prediction at KuaiShou. arXiv preprint arXiv:2407.16357, 2024

168. Gan M, Xiao K. R-RNN: Extracting user recent behavior sequence for click-through rate prediction. IEEE Access, 2019, 7: 111767–111777

169. Lin Q, Zhou W J, Wang Y, Da Q, Chen Q G, Wang B. Sparse attentive memory network for click-through rate prediction with long sequences. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management. 2022, 3312–3321

170. Pi Q, Bian W, Zhou G, Zhu X, Gai K. Practice on long sequential user behavior modeling for click-through rate prediction. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2019, 2671–2679

171. Zhang Y, Chen E, Jin B, Wang H, Hou M, Huang W, Yu R. Clustering based behavior sampling with long sequential data for CTR prediction. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 2195–2200

172. Feng Z, Xie J, Li K, Qin Y, Wang P, Li Q, Yin B, Li X, Lin W, Wang S. Context-based fast recommendation strategy for long user behavior sequence in meituan waimai. In: Companion Proceedings of the ACM on Web Conference 2024. 2024, 355–363

173. Hou R, Yang Z, Ming Y, Lu H, Zheng Z, Chen Y, Zeng Q, Chen M. Cross-domain lifelong sequential modeling for online click-through rate prediction. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024, 5116–5125

174. Yue Z, Wang Y, He Z, Zeng H, McAuley J, Wang D. Linear recurrent units for sequential recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining. 2024, 930–938

175. Liu L, Cai L, Zhang C, Zhao X, Gao J, Wang W, Lv Y, Fan W, Wang Y, He M, others. LinRec: Linear attention mechanism for long-term sequential recommender systems. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023, 289–299

176. Guo T, Li X, Yang H, Liang X, Yuan Y, Hou J, Ke B, Zhang C, He J, Zhang S, others. Query-dominant user interest network for large-scale search ranking. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023, 629–638

177. Cao Y, Zhou X, Feng J, Huang P, Xiao Y, Chen D, Chen S. Sampling is all you need on modeling long-term user behaviors for CTR prediction. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management. 2022, 2974–2983

178. Shen K, Ding X, Zheng Z, Gong Y, Li Q, Liu Z, Zhang G. SEMINAR: Search enhanced multi-modal interest network and approximate retrieval for lifelong sequential recommendation. arXiv preprint arXiv:2407.10714, 2024

179. Ren K, Qin J, Fang Y, Zhang W, Zheng L, Bian W, Zhou G, Xu J, Yu Y, Zhu X, others. Lifelong sequential modeling with personalized memorization for user response prediction. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019, 565–574

180. Dang Y, Yang E, Guo G, Jiang L, Wang X, Xu X, Sun Q, Liu H. Uniform sequence better: Time interval aware data augmentation for sequential recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2023, 4225–4232

181. Zhang S, Chen L, Wang C, Li S, Xiong H. Temporal graph contrastive learning for sequential recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2024, 9359–9367

182. Yin M, Wang H, Guo W, Liu Y, Zhang S, Zhao S, Lian D, Chen E. Dataset regeneration for sequential recommendation. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024, 3954–3965

183. Fan Z, Liu Z, Peng H, Yu P S. Mutual Wasserstein discrepancy minimization for sequential recommendation. In: Proceedings of the ACM Web Conference 2023. 2023, 1375–1385

184. Xiao J, Pan W, Ming Z. A generic behavior-aware data augmentation framework for sequential recommendation. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024, 1578–1588

185. Qiu R, Huang Z, Yin H, Wang Z. Contrastive learning for representation degeneration problem in sequential recommendation. In: Proceedings of the 15th ACM International Conference on Web Search and Data Mining. 2022, 813–823

186. Han Y, Wang H, Wang K, Wu L, Li Z, Guo W, Liu Y, Lian D, Chen E. Efficient noise-decoupling for multi-behavior sequential recommendation. In: Proceedings

of the ACM on Web Conference 2024. 2024, 3297–3306

187. Liu Z, Fan Z, Wang Y, Yu P S. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training Transformer. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, 1608–1612

188. Klenitskiy A, Volodkevich A, Pembek A, Vasilev A. Does it look sequential? an analysis of datasets for evaluation of sequential recommendations. arXiv preprint arXiv:2408.12008, 2024

189. Sun A. On challenges of evaluating recommender systems in an offline setting. In: Proceedings of the 17th ACM Conference on Recommender Systems. 2023, 1284–1285