

Multi-Behavior Recommender Systems: A Survey

Kyungho Kim¹, Sunwoo Kim¹, Geon Lee¹, Jinhong Jung², and Kijung Shin¹(✉)

¹ Kim Jaechul Graduate School of AI, KAIST, Seoul, Republic of Korea
{kkyungho, kswoo97, geonlee0325, kijungs}@kaist.ac.kr

² School of Software, Soongsil University, Seoul, Republic of Korea
jinhong@ssu.ac.kr

Abstract. Traditional recommender systems primarily rely on a single type of user-item interaction, such as item purchases or ratings, to predict user preferences. However, in real-world scenarios, users engage in a variety of behaviors, such as clicking on items or adding them to carts, offering richer insights into their interests. Multi-behavior recommender systems leverage these diverse interactions to enhance recommendation quality, and research on this topic has grown rapidly in recent years. This survey provides a timely review of multi-behavior recommender systems, focusing on three key steps: (1) **Data Modeling**: representing multi-behaviors at the input level, (2) **Encoding**: transforming these inputs into vector representations (i.e., embeddings), and (3) **Training**: optimizing machine-learning models. We systematically categorize existing multi-behavior recommender systems based on the commonalities and differences in their approaches across the above steps. Additionally, we discuss promising future directions for advancing multi-behavior recommender systems.

Keywords: Recommender systems · Multi-behavior recommendation.

1 Introduction

With advancements in web applications, such as e-commerce and streaming platforms, users can conveniently purchase a wide variety of items online. Amid the vast array of available items, users often struggle to find those that best meet their needs. Recommender systems address this challenge by automatically identifying suitable items, including those users may not have considered. In such web applications, users often engage through diverse behaviors beyond simply purchasing or rating items. Actions, such as clicking on items, adding items to shopping carts, and saving items to wish lists, all provide valuable information about user preferences. Traditional recommender systems that focus solely on a single type of interaction may miss these nuanced insights.

Multi-behavior recommender systems seek to harness this spectrum of user interactions to improve the accuracy and relevance of recommendations. By integrating diverse behavioral data, these systems can capture nuanced user intents

and provide more comprehensive personalization. Interest in multi-behavior recommendation has surged in recent years³, reflecting the complex decision-making processes of users and the need for more sophisticated methods to interpret them.

In this survey, we provide an extensive review of multi-behavior recommendation techniques, covering a range of modeling strategies and frameworks. We emphasize the critical role of incorporating multiple user behaviors to improve recommendation accuracy and examine the methodologies used to facilitate this integration. Specifically, we examine three key steps of multi-behavior recommendation: data modeling, encoding, and training. For each step, we present a systematic categorization of existing approaches, focusing on how they leverage multiple behaviors.

Step 1: Data Modeling. Effective data modeling is fundamental for capturing the relationships between multiple user behaviors in recommender systems.

- **View-Specific Graphs:** This approach models each behavior type separately, capturing behavior-specific characteristics and interactions.
- **View-Unified Graph:** By integrating multiple behavior types into a single graph, this approach comprehensively represents user-item interactions.
- **View-Unified Sequences:** By incorporating the temporal order of user behaviors, this approach captures the dynamics of user interactions over time, reflecting evolving preferences.

Step 2: Encoding. Encoding frameworks transform modeled data into user and item representations, which serve as the basis for providing recommendations.

- **Parallel Encoding:** This approach processes each behavior separately or simultaneously, capturing behavior-specific features in parallel.
- **Sequential Encoding:** This approach captures dynamic dependencies across interactions, integrating the progression of user behaviors.

Step 3: Training. Training, the process of optimizing multi-behavior recommender systems, can be guided by various training objectives.

- **Main Training Objectives:** These directly target improving recommendation accuracy, and they are often based on sampling strategies.
- **Auxiliary Training Objectives:** These improve the quality of latent features through auxiliary tasks and/or self-supervised learning techniques.

Comparison with Existing Surveys. While multi-behavior information has been leveraged by recommender systems across various settings and domains, the existing survey [4] focuses on *multi-behavior sequential recommendation*, particularly methods that account for the temporal order of user interactions over time.

³ The number of peer-reviewed publications on this topic increased from 37 in 2019-2020 to 79 in 2021-2022, and reached 164 in 2023-2024. These counts include research articles with "multi-behavior recommendation" or "multi-relation(al) recommendation" in their titles, abstracts, and/or keywords.

By contrast, our survey broadens the scope by offering a more comprehensive exploration of diverse techniques across multiple contexts, with a focus on data modeling, encoding frameworks, and training objectives.

2 Preliminaries for Multi-behavior Recommendation

2.1 Task Description

Overview. The multi-behavior recommendation aims to predict a target behavior (e.g., purchases, likes) by leveraging interactions from both the target behavior itself and auxiliary behaviors (e.g., clicks, add-to-cart, collections, neutral actions, dislikes) (see Figure 1). Since auxiliary behaviors provide valuable signals about potential target behaviors, effectively integrating them is essential for improving recommendation quality.⁴

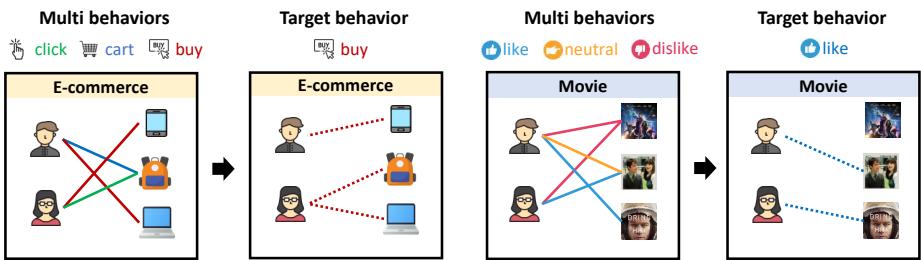


Fig. 1: Multi-behavior recommendation in two example domains.

Key Steps in Multi-behavior Recommendation. We outline the key components of multi-behavior recommendation systems in three main steps.

- S1. Data Modeling:** In this step, multi-behavior interactions are represented using specific data structures, such as graphs or sequences, to capture the distinctive characteristics of multi-behaviors or synergy of them (Section 3).
- S2. Encoding:** Next, the modeled multi-behaviors are transformed into vector representations (i.e., embeddings) to effectively capture meaningful behavioral patterns (Section 4).
- S3. Training:** Finally, the parameters of multi-behavior recommender systems are optimized to enhance their ability to learn and utilize multi-behavior information effectively (Section 5).

⁴ Compared to a method that relies solely on target behavior (LightGCN [13]), state-of-the-art multi-behavior recommender systems (spec., MULE [19]) achieve up to a 463% performance gain in terms of the NDCG@10 metric.

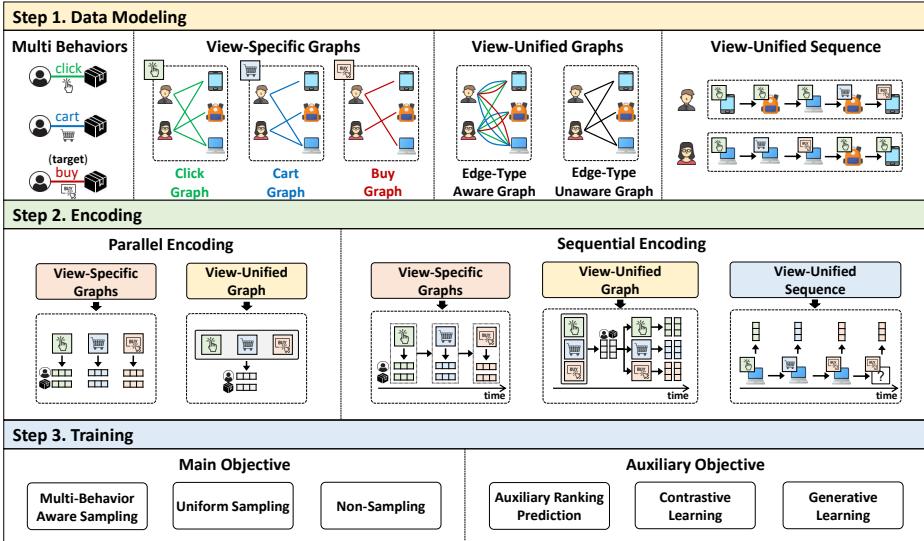


Fig. 2: Overview of key components in multi-behavior recommendation.

We provide an in-depth analysis of current methodologies employed in each of the three phases of multi-behavior recommendation, specifically focusing on how existing methods leverage multi-behavior information. Figure 2 depicts the comprehensive workflow of multi-behavior recommendation, and Table 1 summarizes the key features of existing research in this domain.

2.2 Benchmark Datasets for Multi-behavior Recommendation

Datasets are essential for evaluating multi-behavior recommender systems, enabling researchers to analyze their characteristics and identify effective methods for specific domains. In Table 2, we summarize popular benchmark datasets for multi-behavior recommendation.⁵

Dataset Categorization. The benchmark datasets are categorized into two groups: **without explicit feedback** and **with explicit feedback**. Explicit feedback, such as likes or dislikes [17], directly indicates user preferences.

The first category, without explicit feedback, encompasses datasets based solely on implicit feedback, which does not directly reveal user preferences. Examples include datasets containing clicks, cart additions, and purchase histories, commonly found in e-commerce platforms.

⁵ Due to inconsistencies in dataset statistics caused by variations in preprocessing steps, we include the largest reported number of interactions of each dataset.

The second category, with explicit feedback, includes datasets where user preferences are directly indicated, such as like or dislike histories. This category is common in streaming service platforms.

Dataset Summaries. We present a summary of datasets widely used in multi-behavior recommendation, focusing on interaction types and target behaviors.

Datasets without explicit feedback:

- **Tmall**⁶ is a general e-commerce dataset from Alibaba, containing four types of user behaviors: click, collect, cart, and purchase.
- **Taobao**⁷ is a general e-commerce dataset from Alibaba, containing three types of user behaviors: click, cart, and purchase.
- **Beibei**⁸ is a baby-supply e-commerce dataset from BeiBei, containing three types of user behaviors: click, cart, and purchase.
- **JData**⁹ is a general e-commerce dataset from JD, containing four types of user behaviors: click, collect, cart, and purchase.
- **Online Reatail**¹⁰ is a general e-commerce dataset with four types of user behaviors: click, collect (add-to-favorite), cart, and purchase.
- **IJCAI**¹¹ is a general e-commerce dataset used for an IJCAI competition. It has four types of behaviors: click, add-to-favorite, add-to-cart, and purchase.

In all the above datasets, ‘purchase’ is typically considered the target behavior.

Datasets with explicit feedback:

- **Yelp**¹² is a business-review dataset containing four types of user behaviors: dislike, neutral, like, and tip (i.e., user-provided feedback). Ratings (r) are categorized into three groups: dislike ($r \leq 2$), neutral ($2 < r < 4$), and like ($r \geq 4$), with like serving as the target behavior.
- **ML10M**¹³ is a movie rating dataset from MovieLens, categorizing user behaviors based on ratings into three groups: dislike ($r \leq 2$), neutral ($2 < r < 4$), and like ($r \geq 4$), with like serving as the target.

⁶ <https://tianchi.aliyun.com/dataset/140281>

⁷ <https://tianchi.aliyun.com/dataset/649>

⁸ <https://www.beibei.net/>

⁹ <https://global.jd.com/>

¹⁰ <https://github.com/akaxlh/KHGT>

¹¹ <https://ijcai-15.org/repeat-buyers-prediction-competition/>

¹² <https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset>

¹³ <https://grouplens.org/datasets/movielens/10m/>

Table 1: Comparison of multi-behavior recommender systems.

Model	Modeling				Encoding			Objective		
	View-Spec. Graphs	View-Uni. Graph	View-Uni. Sequences	Others	Parallel	Sequential	Others	Aux. Rank.	Contrastive	Generative
CMF [32]	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗
MR-BPR [18]	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗
MC-BPR [24]	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗
BPRH [31]	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗
VALS [7]	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗
BINN [21]	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗
NMTR [9]	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗
DIPN [11]	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗
MRMN [60]	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗
MATN [40]	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗
MBGCN [15]	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗
EHCF [2]	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗
MGNN [58]	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗
GMNR [39]	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗
MB-GMN [42]	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗
HMG-CR [49]	✗	✗	✗	✓	✓	✗	✗	✓	✓	✗
GHCF [1]	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗
KHGT [41]	✗	✓	✗	✓	✓	✗	✗	✓	✗	✗
S-MBRec [10]	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗
CML [36]	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗
MB-STR [55]	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓
MBHT [51]	✗	✗	✓	✓	✗	✗	✗	✗	✗	✓
MMCLR [52]	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗
NextIP [25]	✗	✓	✗	✗	✗	✓	✓	✗	✓	✓
CKML [28]	✗	✓	✗	✓	✓	✗	✗	✓	✗	✗
CRGCN [28]	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗
MB-CGCN [5]	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗
PKEF [29]	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗
HPMR [30]	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗
CHCF [26]	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗
MB-HGCN [47]	✓	✓	✗	✗	✗	✓	✓	✓	✓	✗
PBAT [33]	✗	✗	✓	✗	✗	✓	✓	✓	✓	✗
KMVLRL [45]	✓	✗	✗	✗	✓	✓	✗	✗	✓	✗
MBA [43]	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗
MBSSL [44]	✓	✗	✗	✗	✗	✓	✗	✓	✓	✗
BCIPM [48]	✓	✗	✗	✗	✗	✓	✗	✓	✓	✗
END4Rec [12]	✗	✗	✓	✗	✗	✓	✓	✗	✓	✗
MBGen [23]	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓
MULE [19]	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗
DA-GCN [61]	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗
HMAR [8]	✗	✗	✓	✗	✗	✓	✓	✓	✗	✗
PO-GCN [59]	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗

Table 2: Summary of datasets used for multi-behavior recommendation. Behaviors providing implicit feedback are highlighted in **bold**, while those providing explicit feedback are underlined. Target behaviors are denoted by \dagger .

Dataset	Field	Types of Behaviors	# of Interactions
Tmall	E-commerce	Click , Collect, Cart, Purchase \dagger	2.3×10^6
Taobao	E-commerce	Click , Cart, Purchase \dagger	7.6×10^6
Beibei	E-commerce	Click , Cart, Purchase \dagger	3.5×10^6
JData	E-commerce	Click , Collect, Cart, Purchase \dagger	2.2×10^6
Online Retail	E-commerce	Click , Collect, Cart, Purchase \dagger	6.4×10^7
IJCAI	E-commerce	Click , Collect, Cart, Purchase \dagger	3.6×10^7
Yelp	Business Reviews	Tip , <u>Dislike</u> , <u>Neutral</u> , <u>Like</u> \dagger	1.4×10^6
ML10M	Movie Ratings	<u>Dislike</u> , <u>Neutral</u> , <u>Like</u> \dagger	9.9×10^6

3 Step 1. Data Modeling

In this and the next two sections, we review recommender systems, focusing on three key steps: **data modeling**, **encoding**, and **training**.

The initial step, data modeling, represents multi-behavior interactions as input-level data structures, such as graphs or sequences. The key challenge of this step lies in how to express rich and diverse interactions effectively. Broadly, there are three categories: (a) **view-specific graphs**, which model each behavior with distinct graphs; (b) **view-unified graph**, which models various behaviors with a single unified graph; and (c) **view-unified sequences**, which model all behaviors of each user with a single sequence. We describe each category below.

3.1 View-Specific Graphs

The view-specific graphs modeling approach represents each type of user interaction as a separate graph.¹⁴ This method independently models the relationships for each behavior, aiming to preserve the unique characteristics of each interaction type. For instance, as illustrated in Figure 2, clicks, add-to-cart actions, and purchases can be expressed as three distinct graphs, each corresponding to a specific behavior type. View-specific graphs can be further classified depending on whether the order between different behaviors is considered.

Non-Ordered View-Specific Graphs. Many studies emphasize the importance of preserving the unique information of each behavior. To achieve this, view-specific graphs are processed in parallel, without considering any order between them [32, 24, 18, 7, 9, 2, 10, 30, 29, 35, 60, 43, 58, 26]. This parallel modeling ensures that the unique properties of each behavior are captured effectively.

¹⁴ We consider single-matrix representation as a type of single-graph representation.

Ordered View-Specific Graphs. Some studies [9,46,5] incorporate an inherent cascading order of user behaviors, such as first clicking an item, then adding it to a cart, and eventually purchasing it. To capture the progression of user behaviors, the view-specific graphs are assigned an order that aligns with the behavior sequence. Later, recommender systems encode these graphs based on the assigned order, which will be elaborated on in Section 4.2.

3.2 View-Unified Graph

The view-unified graph modeling approach consolidates diverse user behaviors into a single graph, capturing their interactions and synergies. This approach can be further categorized into two sub-approaches based on how they handle different edge types within the graph.

Edge-Type-Unaware Unified Graph. The edge-type-unaware unified graph approach aggregates all user behaviors into a single homogeneous graph, treating them as identical types of relationships between users and items [48,47,19]. This simple approach offers several computational advantages and enables the direct extension of the well-known homogeneous graph representation technique (e.g., renormalization trick [16]).

Edge-Type-Aware Unified Graph. The edge-type-aware unified graph approach preserves the type of each edge by leveraging a heterogeneous graph representation [15,40,58,42,39,1]. For instance, each edge contains information on whether the corresponding edge expresses purchase or click. By distinguishing edge types, the graph can effectively capture the unique properties of each behavior while exhibiting the synergy between different interactions at the same time. Upon this heterogeneous graph, message-passing techniques specifically tailored to handle multiple edge types (e.g., HetGNN [57]) are leveraged, which will be elaborated on in Section 4.

3.3 View-Unified Sequences

The view-unified sequence approach is distinct from the previous graph-based modeling techniques in that it explicitly incorporates the sequential nature of user interactions. Unlike methods that ignore the temporal order of user-item interactions, this modeling aims to capture the dynamics of user behaviors over time. This approach leverages a unified sequence that includes all types of behaviors, considering both the order in which they occurred and the specific type of interaction (i.e., behavior). In this method, the view-unified sequence is used as an input to predict the next item for a target behavior [21,11,55,23,51,25,33,8,52,12]. From a behavior-aware user-item interaction sequence, the model gains a richer understanding of user intentions, considering both the temporal dependencies and the distinct characteristics of each behavior.

3.4 Other Representations

In addition to these primary categories, several advanced techniques exploit multi-behavior signals through unique (hyper)graph representations. HMG-CR [49] constructs a hyper meta-graph using hyper meta-paths, which logically combine multiple meta-path schemas to link nodes in heterogeneous graphs. For richer semantic integration, KGHT [41], CKML [28], and KMVL [45] employ knowledge graphs to model item-item relations by leveraging external knowledge with user-item interactions. DA-GCN [61] introduces personalized directed acyclic behavior graphs, incorporating all observed behavior paths to better model behavior dependencies. MBHT [51] employs item-wise hypergraphs, capturing item semantic dependencies and user-personalized multi-behavior dependencies for comprehensive interaction modeling. PO-GCN [59] constructs a unified weighted graph with a graded partial order calculated by a rank function.

4 Step 2. Encoding

Once multi-behavior data is modeled, the encoding step transforms the structured data into meaningful representations that can be effectively used for recommendation. In this section, we categorize encoding techniques into (a) **parallel encoding** and (b) **sequential encoding**. For each category, we provide the underlying intuition, corresponding frameworks, and representative examples.

4.1 Parallel Encoding

The intuition behind parallel encoding is to preserve the unique characteristics of each behavior while enabling collaborative learning across behaviors (either implicitly or explicitly). It assumes no specific order among behaviors and can process them either independently through separate view-specific graphs or collectively by integrating them into a unified graph that distinguishes between interaction types. Existing parallel encoding frameworks can be further categorized into (a) encoding for non-ordered view-specific graphs and (b) encoding for edge-type-aware unified graphs.

Encoding Non-ordered View-Specific Graphs. In this framework, encoding is performed independently for each behavior-specific graph, ensuring that the unique characteristics of each behavior are preserved without interference from others. Various encoding techniques have been employed to learn from each view-specific graph [32,24,10,30,29,35]. For example, CMF [32] employs matrix factorization, while more recent methods, such as S-MBRec [10] and PKEF [29], leverage (homogeneous) graph neural networks (e.g., GCN [16] and LightGCN [13]). These encoding methods facilitate collaborative learning within each behavior while typically leveraging shared initial embeddings across behaviors. The shared initial embeddings enable implicit information sharing among different interaction types.

Encoding an Edge-Type-Aware Unified Graph. In this framework, interactions from multiple behaviors are integrated into a unified graph, while distinguishing which behavior each interaction originates from [15,40,58,42,39,1]. The key challenge in encoding such an edge-type-aware unified graph lies in effectively capturing the inter-dependencies between interactions from different behaviors. For example, MATN [40] employs a Transformer network with embeddings from multiple behaviors as input. Specifically, by leveraging shared key and memory slots, MATN captures cross-behavior dependencies and refines type-specific contextual embeddings. MBGCN [15] applies heterogeneous graph neural networks to propagate information in a unified heterogeneous graph, effectively distinguishing behavior types through edge-type-aware propagation.

4.2 Sequential Encoding

Sequential encoding aims to capture temporal dependencies in multi-behavior interactions. It either assumes an inherent sequence in behavior interactions or leverages explicit temporal information to better understand these dynamics. Existing methods within this category can be further divided into (a) **encoding of ordered view-specific graphs**, (b) **encoding of an edge-type-unaware unified graph**, and (c) **encoding of a view-unified sequence**.

Encoding Ordered View-Specific Graphs. This framework assumes a specific order of behaviors (e.g., click → cart → purchase). Each behavior is modeled as a separate graph, with temporal dependencies explicitly captured to reflect this sequence [9,46,5]. Typically, the output from the previous behavior is used as input to encode the next behavior. Each view-specific graph can be encoded using (neural) matrix factorization [9] or homogeneous graph neural networks [46]. Residual connections across behaviors [46] or linear transformations between behaviors [5] have been used to enhance this approach.

Encoding an Edge-Type-Unaware Unified Graph. This approach leverages an edge-type-unaware unified graph to generate initial embeddings, which are subsequently used as inputs for encoding behavior-specific information [48,47,19]. BIPN [48], for instance, uses an edge-type-unaware unified graph to learn general representations and propagates them into a preference filtering network implemented as a simple MLP with three filtering layers to infer user preference. MULE [19] obtains general representations from an edge-type-unaware unified graph and then propagates them into subgraphs that distinguish (a) auxiliary behaviors intersecting with the target behavior from (b) those that do not. After that, it employs an attention mechanism to discern whether each interaction genuinely indicates user interest.

Encoding View-Unified Sequences. To encode view-unified sequences, it is important to capture temporal dependencies between behaviors of multiple

types. Various sequential models have been employed [21,11,55,23,51,33,8,52]. For example, DIPN [11] uses bidirectional GRUs [6] combined with hierarchical attention to capture intra-sequence and inter-sequence dependencies, while MMCLR [52] employs Bert4Rec [34] to learn behavior-specific representations over time. Transformer-based models, such as MB-STR [55] and MBGen [23], use self-attention mechanisms to model heterogeneous dependencies and fine-grained sequential patterns, leading to a deeper understanding of user interactions. End4Rec [12] utilizes Fourier-transformation-based modules to capture behavior patterns effectively, incorporating denoising mechanisms to effectively filter out noise from user behavior data.

4.3 Others

Many studies develop unique message-passing frameworks and specialized encoding frameworks to capture the intricacies of multi-behavior interactions. EHCF [2] uses a transfer mechanism that projects predictions of preceding behaviors (e.g., view) to subsequent behaviors (e.g., purchase) through translation in the embedding space, capturing the inter-dependencies between behavior types. HMG-CR [49] utilizes GNNs to encode hyper meta-graphs that represent user behavior patterns and capture both semantic relations and topological structures. KHGT [41] employs a hierarchical graph transformer architecture that utilizes graph-based message passing combined with self-attention mechanisms and incorporates temporal dynamics through time-aware embeddings. DA-GCN [61] introduces a directed acyclic graph-based message-passing framework that propagates messages across user-specific and item-specific behavior graphs, refining the embeddings based on cross-behavior dependencies. EIDP [3] introduces a dual-path Transformer-based framework that distinguishes explicit and implicit multi-behavior relationships. The explicit modeling path captures intra-behavior dynamics within each behavior type, while the implicit modeling path focuses on inter-behavior collaborations across behavior types.

4.4 Discussions

Each of these encoding frameworks presents a unique and effective method for encoding user behaviors in multi-behavior recommender systems. Parallel encoding allows for the detailed modeling of behavior-specific patterns by processing each interaction type separately. However, it does not account for the relationships between behaviors, potentially missing the inter-behavior dependencies. Sequential encoding, in contrast, captures the evolving nature of user interactions, making it well-suited for modeling temporal dynamics and providing tailored recommendations. Its drawback lies in the difficulty of integrating a comprehensive view of user preferences across all behaviors.

5 Step 3. Training

In multi-behavior recommender systems, designing an effective training objective is crucial for ensuring recommendation relevance. We categorize training objectives in existing works into (a) **main objectives**, which directly aim to optimize recommendation performance, and (b) **auxiliary objectives**, which indirectly enhance performance by leveraging additional information.

5.1 Main Objectives

The main objective in multi-behavior recommender systems is to accurately predict the target behavior, which is the primary goal of the recommendation task. Achieving this requires designing an effective loss function that guides the model to distinguish between positive and negative samples, ensuring that the scores for positive samples are higher than those for negative samples.¹⁵

Generally, positive samples are straightforward to define, i.e., they are the items with which users have interacted under the target behavior. However, identifying suitable negative samples is more challenging, especially since we lack explicit negative feedback (e.g., dislikes) in many multi-behavior recommendation scenarios. Therefore, selecting proper negative samples is critical to effectively learning user preferences. To select proper negative samples, the following approaches have been studied.

Multi-Behavior-Aware Sampling. Early works [18,24,31,21] on multi-behavior recommendation leverage this strategy to select negative samples that are more informative. For instance, MC-BPR [24] proposes a negative sampling strategy tailored for multi-behavior feedback, assigning different sampling probabilities for distinct cases. In particular, it assigns different sampling probabilities between items that have been viewed but not purchased and those that have not been viewed at all. BPRH [31] introduces an adaptive sampler that considers the co-occurrence of multiple behavior types.

Uniform Sampling. In this approach, negative samples for each user are uniformly selected from items with which the user has not interacted under the target behavior. Many works adopt this uniform sampling due to its simplicity and efficiency [60,40,15,58,42,39,49,41,10,35,51,52,46,5,29,47,33,45,19,48,61,8].

Non-Sampling. Non-sampling-based loss functions eliminate the need for negative sampling, particularly in the context of heterogeneous interactions. This approach considers all items that a user has not interacted with as negative samples. EHCf [2] proposes a non-sampling-based loss function derived from a

¹⁵ Positive samples (or items) are those that users have interacted with, whereas negative items are those they have not.

weighted regression loss [14]. This approach captures multiple behaviors by considering all interactions, rather than relying on sampling. It effectively addresses data sparsity and improves representation quality by utilizing all available information. Other works [1,30,26,44] have also explored this strategy to effectively consider multiple types of relations.

Discussions. Multi-behavior-aware sampling leverages auxiliary behavioral information to select more informative negative samples, enhancing the model’s ability to differentiate among users’ various preferences and capture complex user preferences. Uniform sampling offers simplicity and efficiency but may include less informative negatives or false negatives. Non-sampling-based loss functions eliminate negative sampling procedures, and they can be particularly beneficial in scenarios with data sparsity and heterogeneous interactions.

5.2 Auxiliary Objectives

To further improve model performance and representation quality, especially when data for the target behavior is sparse, many multi-behavior recommender systems incorporate auxiliary objectives. These objectives leverage additional information from auxiliary behaviors to enhance representation learning and improve model robustness. Broadly, auxiliary objectives can be divided into three categories: (a) **auxiliary ranking prediction**, (b) **contrastive learning**, and (c) **generative learning**.

Auxiliary Ranking Prediction. A widely used auxiliary objective is auxiliary ranking prediction, which applies multi-task learning to predict auxiliary behaviors alongside the target behavior [9,11,40,2,42,58,29,30,1,46,47,23,61,8]. Through the joint learning for both target and auxiliary behaviors, the model captures additional signals embedded in user interactions, allowing it to learn shared patterns and underlying correlations. This joint learning approach leads to improved recommendations by leveraging richer behavioral relationships.

Contrastive Learning. Contrastive learning, which is a widely used self-supervised learning technique, has been applied to various recommender systems [22,53,54,27,50,37,36]. It enhances representation learning by effectively capturing meaningful similarities and differences. Contrastive learning has also been adopted in multi-behavior recommender systems [49,10,35,52,25,44,12]. This approach typically contrasts embeddings derived from auxiliary behaviors with those from the target behavior to uncover meaningful relationships. For example, S-MBRec [10] contrasts target and auxiliary behaviors to capture both their commonality and differences, effectively addressing data sparsity while reducing redundancy in auxiliary signals. CML [35] aligns type-specific behavior embeddings using contrastive learning while capturing user-specific behavior heterogeneity through a meta-contrastive network. MMCLR [52] combines multi-

behavior and multi-view contrastive learning to model both coarse-grained commonalities and fine-grained differences, aligning sequence-based and graph-based user representations while capturing behavior-specific priorities. HMG-CR [49] employs graph contrastive learning to compare hyper meta-graphs, adaptively modeling multi-behavior dependencies in a progressive and structured manner. MBSSL [44] combines inter-behavior and intra-behavior contrastive learning to transfer knowledge from auxiliary behaviors. End4Rec [12] contrasts three types of sequences (i.e., denoised sequences, original sequences, and noise sequences) to decouple noise from meaningful user preferences.

Generative Learning. Generative learning has gained increasing attention in recommendation tasks [20,38,56], including multi-behavior recommender systems. It enhances the ability of recommender systems to capture user-item interactions by training them to predict masked items or behaviors [55,51,25,12,23]. For instance, MB-STR [55] utilizes behavior-aware masked item prediction to reconstruct missing items in the interactions of a user. This approach effectively captures multi-behavior dependencies and sequential patterns while mitigating data sparsity and negative transfer. Similarly, MBGen [23] employs sequence-to-sequence generative learning, which auto-regressively predicts the next behavior and item in a heterogeneous sequence of user interactions.

Others. MBA [43] is trained to maximize the likelihood of observed behavioral data while minimizing the KL divergence between auxiliary and target behavior distributions, enabling effective data denoising and knowledge transfer.

6 Challenges and Future Directions

In this section, we discuss key challenges and future research directions in multi-behavior recommendation.

Data Sparsity and Imbalance. In many real-world scenarios, certain types of behaviors are abundant, while others remain sparse. This imbalance may cause recommender systems to prioritize learning abundant behaviors while failing to capture sparse ones. Advanced self-supervised learning methods that incorporate external knowledge offer a promising direction for addressing these issues.

Scalability and Efficiency. Real-world user behaviors often occur at scale, making scalability essential for multi-behavior recommender systems to handle them effectively. Developing a scalable recommender system along with an efficient training strategy, such as sampling-based techniques, would be a promising research direction.

Temporal Dynamics. User preferences change over time, making it crucial to capture these dynamics for relevant recommendations. A promising research direction involves integrating temporal attention mechanisms or dynamic graph-based methods that adaptively update embeddings to reflect time-sensitive changes in user behavior and the evolving relationships between users.

Interpretability. Deep learning-based recommender systems, especially those leveraging complex architectures, often act as black boxes. Enhancing interpretability to explain how different behaviors influence recommendations is a crucial research direction for fostering user trust and transparency. To achieve this, various techniques, such as attention visualization and counterfactual reasoning, can be leveraged.

Privacy and Ethical Considerations. Detailed user behavioral data raises privacy concerns, making security and ethical compliance essential. A key research direction involves exploring privacy-preserving methods, such as federated learning and differential privacy, to ensure the secure training of multi-behavior recommender systems while minimizing data exposure risks.

7 Conclusions

In this survey, we provide the first comprehensive review of multi-behavior recommender systems. We systematically categorize existing multi-behavior recommendation systems based on their approaches across three key steps: data modeling, encoding, and training. Specifically, we examine how multi-behavior data are modeled in various forms and effectively encoded to capture nuanced relationships across multiple behaviors. Additionally, we investigate the role of primary and auxiliary training objectives in leveraging diverse behavioral signals. Notably, unlike previous surveys, our work covers multi-behavior recommendation across various contexts beyond sequential recommendation.

By fostering a systematic understanding of multi-behavior recommender systems, from early research to recent advancements, this survey aims to aid researchers and practitioners in developing more sophisticated and effective solutions for multi-behavior recommendation.

Acknowledgements. This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00438638, EntireDB2AI: Foundations and Software for Comprehensive Deep Representation Learning and Prediction on Entire Relational Databases, 50%) (No. RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST), 10%). This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00406985, 40%).

References

1. Chen, C., Ma, W., Zhang, M., Wang, Z., He, X., Wang, C., Liu, Y., Ma, S.: Graph heterogeneous multi-relational recommendation. In: AAAI (2021)
2. Chen, C., Zhang, M., Ma, W., Zhang, Y., Liu, Y., Ma, S.: Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In: AAAI (2020)
3. Chen, M., Pan, W., Ming, Z.: Explicit and implicit modeling via dual-path transformer for behavior set-informed sequential recommendation. In: KDD (2024)
4. Chen, X., Li, Z., Pan, W., Ming, Z.: A survey on multi-behavior sequential recommendation. arXiv preprint arXiv:2308.15701 (2023)
5. Cheng, Z., Han, S., Liu, F., Zhu, L., Gao, Z., Peng, Y.: Multi-behavior recommendation with cascading graph convolution networks. In: WWW (2023)
6. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: EMNLP (2014)
7. Ding, J., Yu, G., He, X., Quan, Y., Li, Y., Chua, T.S., Jin, D., Yu, J.: Improving implicit recommender systems with view data. In: IJCAI (2018)
8. Elsayed, S., Rashed, A., Schmidt-Thieme, L.: Hmar: Hierarchical masked attention for multi-behaviour recommendation. In: PAKDD (2024)
9. Gao, C., He, X., Gan, D.: Neural multi-task recommendation from multi-behavior data. In: ICDE (2019)
10. Gu, S., Wang, X., Chi, C., Xiao, D.: Self-supervised graph neural networks for multi-behavior recommendation. In: IJCAI (2022)
11. Guo, L., Hua, L., Jia, R., Zhao, B., Wang, X., Cui, B.: Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior. In: KDD (2019)
12. 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: WWW (2024)
13. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: SIGIR (2020)
14. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: ICDM (2008)
15. Jin, B., Gao, C., He, X., Jin, D., Li, Y.: Multi-behavior recommendation with graph convolutional networks. In: SIGIR (2020)
16. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
17. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: KDD (2008)
18. Kronh-Grimbergh, A., Drumond, L., Freudenthaler, C., Schmidt-Thieme, L.: Multi-relational matrix factorization using bayesian personalized ranking for social network data. In: WSDM (2012)
19. Lee, S., Ko, G., Song, H.J., Jung, J.: Mule: Multi-grained graph learning for multi-behavior recommendation. In: CIKM (2024)
20. Li, C., Xia, L., Ren, X., Ye, Y., Xu, Y., Huang, C.: Graph transformer for recommendation. In: SIGIR (2023)
21. Li, Z., Zhao, H., Liu, Q., Huang, Z., Mei, T., Chen, E.: Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In: KDD (2018)

22. Lin, Z., Tian, C., Hou, Y., Zhao, W.X.: Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In: WWW (2022)
23. Liu, Z., Hou, Y., McAuley, J.: Multi-behavior generative recommendation. In: CIKM (2024)
24. Loni, B., Pagano, R., Larson, M., Hanjalic, A.: Bayesian personalized ranking with multi-channel user feedback. In: RecSys (2016)
25. Luo, J., He, M., Lin, X., Pan, W., Ming, Z.: Dual-task learning for multi-behavior sequential recommendation. In: CIKM (2022)
26. Luo, X., Wu, D., Gu, Y., Chen, C., Liu, L., Ma, J., Zhang, M., Deng, M., Huang, J., Hua, X.S.: Criterion-based heterogeneous collaborative filtering for multi-behavior implicit recommendation. ACM Transactions on Knowledge Discovery from Data **18**(14), 1–26 (2023)
27. Ma, Y., He, Y., Zhang, A., Wang, X., Chua, T.S.: Crosscbr:cross-view contrastive learning for bundle recommendation. In: KDD (2022)
28. Meng, C., Zhao, Z., Guo, W., Zhang, Y., Wu, H., Gao, C., Li, D., Li, X., Tang, R.: Coarse-to-fine knowledge-enhanced multi-interest learning framework for multi-behavior recommendation. ACM Transactions on Information Systems **42**(30), 1–27 (2023)
29. Meng, C., Zhai, C., Yang, Y., Zhang, H., Li, X.: Parallel knowledge enhancement based framework for multi-behavior recommendation. In: CIKM (2023)
30. Meng, C., Zhang, H., Guo, W., Guo, H., Liu, H., Zhang, Y., Zheng, H., Tang, R., Li, X., Zhang, R.: Hierarchical projection enhanced multi-behavior recommendation. In: KDD (2023)
31. Qiu, H., Liu, Y., Guo, G., Sun, Z., Zhang, J., Nguyen, H.T.: Bprh: Bayesian personalized ranking for heterogeneous implicit feedback. Information Sciences **453**, 80–98 (2018)
32. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: KDD (2008)
33. Su, J., Chen, C., Lin, Z., Li, X., Liu, W., Zheng, X.: Personalized behavior-aware transformer for multi-behavior sequential recommendation. In: MM (2023)
34. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: CIKM (2019)
35. Wei, W., Huang, C., Xia, L., Xu, Y., Zhao, J., Yin, D.: Contrastive meta learning with behavior multiplicity for recommendation. In: WSDM (2022)
36. Wei, Y., Wang, X., Li, Q., Nie, L., Li, Y., Li, X., Chua, T.S.: Contrastive learning for cold-start recommendation. In: MM (2021)
37. Wu, J., Fan, W., Chen, J., Liu, S., Li, Q., Tang, K.: Disentangled contrastive learning for social recommendation. In: CIKM (2022)
38. Xia, L., Huang, C., Huang, C., Kangyi, L., Yu, T., Ben, K.: Automated self-supervised learning for recommendation. In: WWW (2023)
39. Xia, L., Huang, C., Xu, Y., Dai, P., Lu, M., Bo, L.: Multi-behavior enhanced recommendation with cross-interaction collaborative relation modeling. In: ICDE (2021)
40. Xia, L., Huang, C., Xu, Y., Dai, P., Zhang, B., Bo, L.: Multiplex behavioral relation learning for recommendation via memory augmented transformer network. In: SIGIR (2020)
41. Xia, L., Huang, C., Xu, Y., Dai, P., Zhang, X., Yang, H., Pei, J., Bo, L.: Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In: AAAI (2021)

42. Xia, L., Xu, Y., Huang, C., Dai, P., Bo, L.: Graph meta network for multi-behavior recommendation. In: SIGIR (2021)
43. Xin, X., Liu, X., Wang, H., Ren, P., Chen, Z., Lei, J., Shi, X., Luo, H., Jose, J.M., de Rijke, M., Ren, Z.: Improving implicit feedback-based recommendation through multi-behavior alignment. In: SIGIR (2023)
44. Xu, J., Wang, C., Wu, C., Song, Y., Zheng, K., Wang, X., Changping, W., Zhou, G., Gai, K.: Multi-behavior self-supervised learning for recommendation. In: SIGIR (2023)
45. Xuan, H., Liu, Y., Li, B., Yin, H.: Knowledge enhancement for contrastive multi-behavior recommendation. In: WSDM (2023)
46. Yan, M., Chen, Z., Gao, C., Sun, Z., Liu, F., Sun, F., Li, H.: Cascading residual graph convolutional network for multi-behavior recommendation. ACM Transactions on Information Systems **1**, 1–24 (2023)
47. Yan, M., Cheng, Z., Sun, J., Sun, F., Peng, Y.: Mb-hgcn: A hierarchical graph convolutional network for multi-behavior recommendation. arXiv preprint arXiv:2306.10679 (2023)
48. Yan, M., Liu, F., Sun, J., Sun, F., Chen, Z., Han, Y.: Behavior-contextualized item preference modeling for multi-behavior recommendation. In: SIGIR (2024)
49. Yang, H., Chen, H., Li, L., Yu, P.S., Xu, G.: Hyper meta-path contrastive learning for multi-behavior recommendation. In: ICDM (2021)
50. Yang, Y., Huang, C., Xia, L., Li, C.: Knowledge graph contrastive learning for recommendation. In: SIGIR (2022)
51. Yang, Y., Huang, C., Xia, L., Liang, Y., Yu, Y., Li, C.: Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In: KDD (2022)
52. Yiqing, W., Ruobing, X., Yongchun, Z., Xiang, A., Xin, C., Xu, Z., Fuzhen, Z., Leyu, L., Qing, H.: Multi-view multi-behavior contrastive learning in recommendation. In: DASFAA (2022)
53. Yu, B., Zhang, R., Chen, W., Fang, J.: Graph neural network based model for multi-behavior session-based recommendation. GeoInformatica **26**(2), 429–447 (2022)
54. Yu, J., Xia, X., Chen, T., Cui, L., Hung, N.Q.V., Yin, H.: Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. IEEE Transactions on Knowledge and Data Engineering **36**(2), 913–926 (2023)
55. Yuan, E., Guo, W., He, Z., Guo, H., Liu, C., Tang, R.: Multi-behavior sequential transformer recommender. In: SIGIR (2022)
56. Zhai, J., Liao, L., Liu, X., Wang, Y., Li, R., Cao, X., Gao, L., Gong, Z., Gu, F., He, M., Lu, Y., Shi, Y.: Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In: ICML (2024)
57. Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V.: Heterogeneous graph neural network. In: KDD (2019)
58. Zhang, W., Mao, J., Cao, Y., Xu, C.: Multiplex graph neural networks for multi-behavior recommendation. In: CIKM (2020)
59. Zhang, Y., Bei, Y., Chen, H., Shen, Q., Yuan, Z., Gong, H., Wang, S., Huang, F., Huang, X.: Multi-behavior collaborative filtering with partial order graph convolutional networks. In: KDD (2024)
60. Zhou, X., Liu, D., Lian, J., Xie, X.: Collaborative metric learning with memory network for multi-relational recommender systems. In: IJCAI (2019)
61. Zhu, X., Lin, F., Zhao, Z., Xu, T., Zhao, X., Yin, Z., Li, X., Chen, E.: Multi-behavior recommendation with personalized directed acyclic behavior graphs. ACM Transactions on Information Systems **43**(1), 1–30 (2024)