JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2021

Behavior Habits Enhanced Intention Learning for Session Based Recommendation

Wenhao Xue, Zhida Qin, Haoyao Zhang, Shixiao Yang, Enjun Du, Shuang Li, *Member, IEEE,* Tianyu Huang, John C.S. Lui, *Fellow, IEEE*

***Abstract*—Multi-behavior Session Based Recommendations (MBSBRs) have achieved remarkable results due to considering behavioral heterogeneity in sessions. Yet most existing works only consider binary or continuous behavior dependencies and aim to predict the next item under the target behavior, neglecting users' inherent behavior habits, resulting in learning inaccurate intentions. To tackle the above issues, we propose a novel Behavior Habits Enhanced Intention Learning framework for Session Based Recommendation (BHSBR) framework. Specifically, we focus on the next item recommendation and design a global item transition graph to learn the behavior-aware semantic relationships between items, in order to mine the underlying similarity between items beyond the session. In addition, we construct a hypergraph to extract the diverse behavior habits of users and break through the limitations of temporal relationships in the session. Compared to the existing works, our behavior habit learning method learns behavior dependencies at the user level, which could capture the user's more accurate long-term intentions and reduce the impact of noise behaviors. Extensive experiments on three datasets demonstrate that the performance of our proposed BHSBR is superior to SOTA. Further ablation experiments fully illustrate the effectiveness of our various modules.**

***Index Terms*—Session based Recommendation, Multi-behavior Recommendation, Graph Neural Network.**

I. Introduction

SESSION-BASED recommendation algorithms aim to make the next-item prediction for anonymous, temporal sessions [1]–[3]. Due to the increasingly improved protection of user privacy in recent years, session-based recommendations have gradually become a hot research area. Traditional methods [4], [5] utilize Markov chains to encode sessions. However, these methods can only predict the latter term based on the previous one, which results in insufficient utilization of all sequence information of the session. To address this issue, some methods [6]–[8] leverage RNN to model sessions and capture temporal relationships in the sequence. Recently, due to the superiority of GNN in capturing structural features, many methods [9]–[11] have introduced GNN into the process of feature learning and achieved sound results.

Whereas, the above methods only consider the macro connectivity between items, without utilizing the complex behavioral information between users and items, and cannot capture fine-grained user preferences. For example, in shopping scenarios, compared to products that users have only browsed, users are clearly more interested in products that are more relevant to those they have added to their shopping cart or have already purchased. Therefore, when considering behavioral information, we are better able to capture the user's intention. Recently, some methods have introduced multi-behavior information into session recommendation. TGT [12] proposed a temporal graph Transformer to capture dynamic user-item interaction patterns by exploring the evolutionary correlations between behaviors. MBHT [13] designed a multi-scale transformer to learn the behavioral order patterns of sequences and utilized hypergraphs to learn cross-behavioral dependencies. EMBSR [14] designed multi graph to aggregate micro-operations in sequences and utilized extended attention mechanisms to learn binary behavior patterns.

Although these methods have achieved delightful results in aggregating sequence representations of behavioral information and mining relationships between behavioral dependencies, there are still some issues that need to be addressed:

- **Failing to learn user-level behavior habits**. Current Multi-Behavior Session-Based Recommendation methods (MBSBR) are limited to learning behavior dependencies at the behavior or item level. Behavior-level methods [15], [16] only model single behavior, and struggle with complex multi-behavior scenarios. Item-level methods [17], though capable of capturing cross-behavior dependencies, often only address binary patterns or aggregate behaviors in continuous timestamps. This limitation means they fail to capture the full spectrum of user behavior habits towards items. The existing approaches do not sufficiently mine user-level dependencies, which are crucial for accurately reflecting user intentions.
- **Predicting the targeted behaviors is not reasonable in session-based recommendations.** Current MBSBR methods often define a target behavior and treat other behaviors as auxiliary, focusing on predicting the next item for that target. However, in reality, users frequently engage in auxiliary behaviors before performing target behavior. This approach can lead to systems recommending items that users

Wenhao Xue, Zhida Qin, Haoyao Zhang, Shixiao Yang and Tianyu Huang are with the school of Computer Science, Beijing Institute of Technology, Beijing 100081, China (email: aqxuewenhao@bit.edu.cn; zanderqin@bit.edu.cn; zhanghaoyao@bit.edu.cn; ysx144_51@bit.edu.cn; huangtianyu@bit.edu.cn).

Enjun Du is with the school of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (email: enjundu666@gmail.com).

Shuang Li is with the school of Artificial Intelligence, Beihang University, Beijing 100191, China (email: shuangliai@buaa.edu.cn).

John C.S. Lui is with the department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T, Hong Kong (email: cslui@cse.cuhk.edu.hk).

have already interacted with rather than new ones, which contradicts the purpose of recommendation systems and may degrade the user experience.

To address the aforementioned issues, we propose a novel Behavior Habits Enhanced Intention Learning framework for Session Based Recommendation (**BHSBR**) framework. First, we focus on achieving the next-item recommendation that users may be interested in, and design an intention learning module for global semantic fusion behavior. This module could efficiently integrate the global item transition relationships and internal behavior temporal information, which further learn the behavior-aware semantic relationships between items. Technically, we construct a global item interaction graph to extract semantic connections between items. Then, we aggregate behavior sequences under macro items and fuse semantic connections with behavioral information through gating mechanisms to capture fine-grained user intentions. Meanwhile, we capture the coarse-grained intent representation contained in the macro item sequence of the session, aggregate it with fine-grained intent, and learn the final intention representation of the user. Second, our approach leverages hypergraphs to capture user behavior habits and deduce dependencies at the user level. To elaborate, we designate items as vertices within the hypergraph and construct hyperedges that encompass all item-related behaviors occurring within a session. This technique enables us to extract user behavior habits from sessions, transcending constraints of time and number, thereby enhancing user intentions and reducing noise within behavioral sequences. Concurrently, to counteract the challenges of gradient explosion and over-smoothing, we add residual connections when constructing the hypergraph. Finally, we design a mutual information fusion module that adaptively fuses the information of the two modules to learn the final representation of the session.

In summary, the main contributions of this paper are as follows:

- We propose the **BHSBR** framework. It captures behavior habits in sessions by constructing hypergraphs to learn user-level behavior dependencies, and innovatively utilizes global semantic relationships and item behavior sequences for fusion to learn session intent representations. We adaptively fuse the two types of information to obtain the final session preference.
- We focus on a more reasonable MBSBR problem, which aims to discover items that users may be interested in rather than simply speculating on items under the target behavior. We construct a global graph to search the entire semantic space combine behavioral information, and learn complex user behavior habits, in order to more accurately capture user intentions.
- We conduct extensive experiments on three real-world datasets, and our model outperforms SOTA, demonstrating the effectiveness of our framework. Meanwhile, further ablation experiments have also verified that each of our modules has its own effect.

The remainder of this paper is organized as follows. Section II briefly reviews current related work. Section III analyzes the data and defines the multi-behavior session-based recommendation. Section IV introduces the framework of BHSBR in detail. Section V analyzes the results of experiments. Finally, Section VI summarizes the paper.

II. Related Work

A. Session-based Recommendation

Due to the lack of user information and temporal interaction, session recommendation algorithms differ from general recommendation paradigms [18], [19]. FPMC [4] combined the advantages of Matrix Factorization (MF) and Markov Chain (MC) to achieve personalized next recommendation. However, the MC-based method can only calculate similarity based on the last item in the session, ignoring the information on the previous items. RNN can effectively solve this problem. Hereby, GRU4Rec [6] first utilized the GRU method to model sessions and learn the temporal preferences of the entire session. Meanwhile, as the attention mechanism demonstrated its superiority in capturing sequence temporal features [20], NARM [7] added an attention layer to GRU4Rec for generating session representations. STAMP [21] considered both the general interest of long-term memory in the session context and the short-term interest of the last click, and utilized attention mechanisms to obtain the final representation. Bert4Rec [22] designed a deep bidirectional self-attention strategy to capture correlations between items. Most of these RNN and attention-based methods focused on capturing temporal features while paying insufficient attention to structural features in sessions.

Recently, more and more methods have introduced GNN into session recommendation. SRGNN [23] was the first to design a gated GNN framework for learning the sequential structure in a session. SGNN-HN [24] designed a star GNN model to capture high-order information that may exist between non-adjacent items in a session, and designed a Highway Network to adaptively learn embedding information from item representations to alleviate the overfitting problem in GNN. The above methods only encoded a single session and did not effectively utilize the global information of the data. In order to fully utilize the information in the dataset, GCE-GNN [25] learned sequence representation by constructing two views, a session graph and a global graph. COTREC [26] introduced contrastive learning into session recommendation, generated labels using different connection relationships, and supervised through contrastive learning strategies to obtain session representations. MCLRec [27] proposed a meta-optimized contrastive learning framework, which uses meta-learning to guide model enhancers in updating, while considering contrastive regularization terms to solve the problems of previous contrastive pairs being difficult to generalize and lacking available information. Furthermore, some methods use the idea of hypergraphs to model sessions. DHCN [28] first modeled session data into hypergraphs to learn session embedding. HIDE [29] mapped an item to multiple embeddings, each corresponding to an intention, and disentangled the intentions from both macro and micro perspectives to achieve prediction.

B. Multi-behavior Recommendation

The multi-behavior recommendation aims to use behavioral data in interactions to encode users and items, capture dynamic preferences in behavior, and improve recommendation performance [30]–[33]. The traditional multi-behavior recommendation algorithm [34] extended the matrix factorization algorithm to achieve multi-behavior data encoding. However, such methods cannot capture higher-order information about users and items at the behavioral level. Most recent methods utilized GNN to propagate the messages contained in behaviors in order to capture complex transformation relationships [35]–[37]. CML [38] and MBSSL [39] combined GNN and self-supervised learning methods to capture dependencies and diversity between behaviors. However, user behavior often contains deep information in chronological order. These methods only considered the behavioral interaction information between users and items and did not delve into the deep-level temporal relationships of behaviors. RIB [40] introduces RNN layers into the model to learn fine-grained behavior sequence representations. MKM-SR [41] designed a multi-task learning framework that combined knowledge graphs with session sequences and behavioral information for loss calculation. PBAT [42] developed a personalized behavior pattern generator to extract unique user behavior patterns and designed a behavior-aware collaborative extractor that utilizes attention mechanisms to extract collaborative relationships in sequences. Although these methods have learned behavioral dependencies, they have not explored deeper behavioral habits and cannot better capture users' personalized behavioral preferences.

III. Preliminary

A. Behavior Habits Definition

Behavior habit is defined as a sequence composed of all behaviors the user taken towards a specific item within a session. For example, in session s_e, if there are browsing, adding to cart, and purchasing operations on item v_i, it is considered that the *behavior habit* for v_i of the user is $[browse \rightarrow carting \rightarrow purchase]$. A *behavior habit* reflects the entire interaction process of a user with an item. These habits may not be consciously aware by users, which are instrumental in assisting in predicting recommendations.

B. Data Analysis

In order to explore the actual behavioral dependencies that exist in the real world, understand how to learn session behavioral dependencies, and explore the rationality of current MBSBR tasks, we conduct data analysis on two datasets in the real world, *Appliances* and *Computers*. These two datasets are detailed in section V-A1. Specifically, we select a session from *Appliances* and count the number of behavior habits in that session. At the same time, we conduct a detailed statistical analysis of the item behavior sequences in each dataset, including all behavior habits in the dataset. In summary, we conduct data analysis from both individual and holistic perspectives, aiming to answer the following two questions:

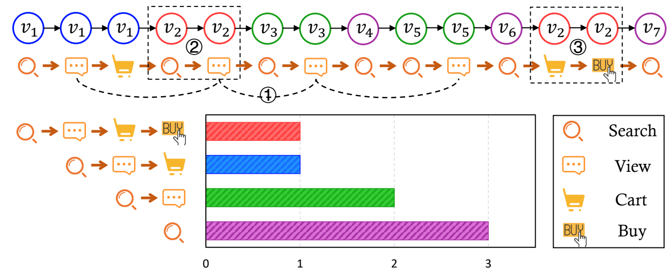

Fig. 1. Counts on all behavioral habits in session #37 in *Appliances* dataset.

- RQ1: Can user behavior habits effectively reflect their intentions? Is learning user-level behavioral dependencies more reasonable than previous methods?
- RQ2: Currently, most MBSBR tasks set a target behavior and set other types of behavior as auxiliary behaviors, aiming to predict the next item under the target behavior. Is this task setting reasonable?

1) RQ1: In order to explore the effects of different behavioral dependency learning methods in existing works, we conduct in-depth research on the session. Specifically, we randomly select one session #37 with multiple items and behaviors in Fig. 1 from the *Appliances* dataset. First, scene ① represents behavior-level methods, which aggregates item v_1, v_2, v_3, and v_5 with "$view$" behavior. Such methods can only learn dependency relationships under different behaviors independently of each other, thereby lacking the capacity to learn complex cross-behavior dependency patterns. Second, scenarios ② and ③ show the learning objects of item-level methods. For item v_2, this method can learn the behavior dependency of $[search \rightarrow view]$ in ②, and the behavior dependency of $[cart \rightarrow buy]$ in ③. However, these methods either only learn binary behavior patterns, or only aggregate different behaviors of items in continuous timestamps, without mining the behavior dependency of $[view \rightarrow cart]$, consequently failing to capture the complete behavior habits of users towards items. Additionally, by examining session #37's behavior habits, we deduce the user's shopping pattern: initiating with searches, followed by reviewing detailed information or comments. Upon thorough evaluation, interested items are added to the cart or purchased. Consequently, capturing user-level behavior habits can more accurately identify which items have a greater impact on user intent, achieving enhancement of user intention.

2) RQ2: To explore the rationality of the MBSBR task, we count the number of all behavior habits in two datasets. Due to the large number of behavioral habits in the dataset, we select 5 typical behavior habits for display as shown in Fig. 2. Usually, the "*buy*" behavior is considered as the target behavior. However, according to our statistics, out of nearly ten million behavioral habits, the number of individual "*buy*" behaviors of users for items in both datasets is 0. Moreover, the total number of behavior habits that include "*buy*" accounts for less than 0.2% of the total. This figure shows the two behavior habits with the highest number of "*buy*" behaviors, and it can be seen that there is a difference of hundreds of

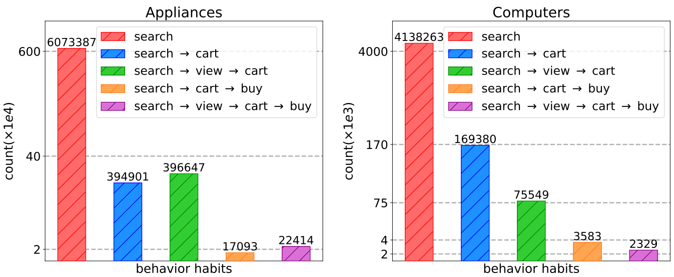

Fig. 2. Five Typical Behavior Habits on Two Datasets.

times compared to the most common [*search*] behavior habit. This phenomenon indicates that in a session if there is a "*buy*" behavior for an item, other behaviors must be taken before that behavior occurs. In this case, if the recommender system still predicts the items under the target behavior, the system will ultimately be more inclined to recommend products that the user has already interacted with, rather than recommending new items to the user. This contradicts the fundamental purpose of a recommendation system. The repetitive recommendation of items that users have already engaged with could potentially diminish the overall satisfaction and engagement of the user. In addition, our results also find that compared to other behaviors, user "*buy*" behavior accounts for very small proportion (only 2% in *Appliances* dataset and 0.3% in *Computers* dataset), such extremely small number of labels is hardly to produce accurate item predictions under the "buy" target behavior. In summary, predicting the recommendation task of items under target behavior is unreasonable on MBSBR.

C. Task Formulation

In this section, we formulate the task of our multi-behavior session-based recommendation.

Task Scenario. Suppose that we have a set of sessions denoted as $\mathcal{S} = \{s_1, s_2, \cdots, s_M\}$, where s_i represents the i-th session and M represents the number of all sessions. For session $s_i = [v_1^{b_{1,1}}, v_1^{b_{1,2}}, v_2^{b_{2,1}}, \cdots, v_n^{b_{n,m}}]$, the element $v_t^{b_{t,m}}$ represents the t-th interacted item in session s_i with behavior $b_{t,m}$, where $b_{t,m}$ represents the m-th behavior performed by item v_t, indicating a single behavior user can perform, such as searching, buying, etc.

Prediction Process. Our recommendation system is dedicated to making the next-item prediction for the session. However, due to the diversity of behavioral patterns, there may be multiple consecutive identical items in a session. For example, suppose a session $s_{eg} = [v_1^{b_{1,1}}, v_1^{b_{1,2}}, v_2^{b_{2,1}}, \cdots, v_n^{b_{n,1}}, v_n^{b_{n,2}}]$. Its last two items are all v_n, with only different behaviors being performed. In this case, the predicted next item is likely to still be v_n, which leads to information leakage. To avoid this situation, we define consecutive identical items in a session as a macro item and reconstruct new sessions containing macro items for the original session set $\mathcal{S}$. For session s_{eg} in the previous example, the reconstructed session is $s_{new} = [v_1, v_2, \cdots, v_n]$, and every macro item v_i has a corresponding behavior sequence $h_i = [b_{i,1}, b_{i,2}, \cdots, b_{i,l}]$, where $b_{i,j}$ means the j-th behavior targeting v_i and l represents the length of the behavior sequence. Our goal is to utilize the above information to predict the next macro item v_{n+1}. We formally set out the problem as follows:

- **Inputs:** The original session $s = [v_1^{b_{1,1}}, v_1^{b_{1,2}}, v_2^{b_{2,1}}, \cdots, v_n^{b_{n,m}}]$, the reconstructed session $s' = [v_1, v_2, \cdots, v_n]$ and the behavior sequence list $H = [h_1, h_2, \cdots, h_n]$ related to the macro items in s'.
- **Outputs:** The predicted next macro item v_{n+1} for this session from the item set.

IV. Methodology

A. Workflow Overview

Our proposed framework of **BHSBR** is illustrated in Fig. 3. We design a globally directed graph to extract semantic relationships in the item space, and utilize GRU to capture the behavioral sequential relationship of each macro item, in order to obtain the intention representation of behavior-aware fused semantics. Meanwhile, for the input session, we construct a hypergraph to capture the behavior habits in the session and update the item representation through a hypergraph neural network to enhance user intention. Finally, we fuse the two types of mutual information to obtain the final session representation.

B. Intention Learning

1) Global Item Transition Capturing: Currently, most multi-behavior recommendation methods only encode the items in the current session, ignoring the global complex high-order item transition relationships, which is beneficial to learning item embedding in the session. We construct a global item interaction graph to aggregate global-level neighborhood information.

Define item set as $\mathcal{V}$ and the number of all items is N. If in the session set $\mathcal{S}$ there exist $v_i, v_j \in \mathcal{V}$ such that $v_i \rightarrow v_j$, which means that the user interacts with v_j after interacting with v_i. We define v_j as the subsequent neighbor of v_i, and thus obtain the set of all items' subsequent neighbors $\mathcal{N}$. So the global graph we construct can be represented as $\mathbf{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{E} = \{(v_i, v_j) | v_i \in \mathcal{V}, v_j \in \mathcal{N}(v_i)\}$ represents the interactive edge set at global-level and $\mathcal{N}(v_i)$ is the subsequent neighbors set of item v_i. According to $\mathbf{G}$, we generate a normalized adjacency matrix $\mathbf{A}^g \in \mathbb{R}^{N \times N}$ as follows:

$$a_{ij}^g = \frac{g_{ij}}{\sum_{k=1}^{N} g_{ik}}, \quad (1)$$

where a_{ij}^g and g_{ij} respectively represent elements from $\mathbf{A}^g$ and $\mathbf{G}$. Whereupon we design a simplified graph convolution layer to globally encode the items:

$$\mathbf{E}_g^{(l+1)} = \mathbf{D}^{-1} \mathbf{A}^g \mathbf{E}_g^{(l)} \mathbf{W}^{(l)}, \quad (2)$$

where $\mathbf{E}_g^{(l)} \in \mathbb{R}^{N \times d}$ and $\mathbf{W}^{(l)} \in \mathbb{R}^{d \times d}$ means the item embedding and parameter matrix in the l-th layer respectively, and d represents the embedding dimension size as well as $\mathbf{D} \in \mathbb{R}^{N \times N}$ is the degree matrix. After L_g-layer propagation, we obtained an item representation $\mathbf{E}_g^{(L_g)}$ that integrates global high-order neighborhood information.

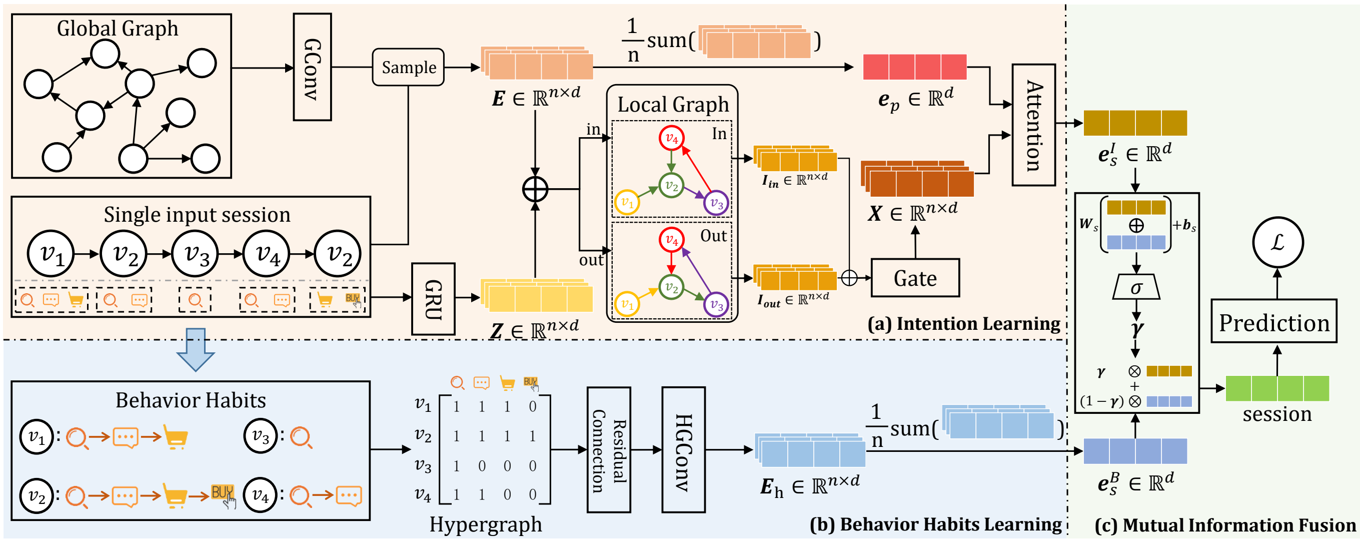

Fig. 3. The overview of the proposed BHSBR.

2) Macro Item Representation Learning: According to section III-C, for original session $s = [v_1^{b_{1,1}}, v_1^{b_{1,2}}, v_2^{b_{2,1}}, \cdots, v_n^{b_{n,m}}]$, we can obtain the reconstructed session $s' = [v_1, v_2, \cdots, v_n]$ and behavior sequence list $H = [h_1, h_2, \cdots, h_n]$, where $h_i = [b_{i,1}, b_{i,2}, \cdots, b_{i,l}]$ represents the behavior sequence of macro item v_i. After obtaining the item embedding representation that aggregates global neighborhood information, we design a macro item representation generator to capture the personalized features of each macro item in the session.

In order to extract the behavioral sequence characteristics of macro item v_i, we send its corresponding behavior sequence h_i to GRU:

$$\mathbf{z}_{i,j} = GRU(\mathbf{e}_{i,j}^b, \mathbf{z}_{i,j-1}, \Phi_{GRU}), \tag{3}$$

where $\mathbf{z}_{i,j} \in \mathbb{R}^d$ and $\mathbf{e}_{i,j}^b \in \mathbb{R}^d$ indicate the hidden state and the embedding representation of the behavior $b_{i,j}$ respectively, and Φ_{GRU} is all parameters of GRU. After processing by GRU, the hidden state $\mathbf{z}_{i,l}$ of the last behavior in h_i contains the aggregated information throughout the entire behavior sequence. As a consequence, we consider it as the behavioral feature $\mathbf{z}_i$ of the macro item v_i and we can get the behavior temporal feature of all macro items $\mathbf{Z} \in \mathbb{R}^{n\times d}$ in the session:

$$\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n]. \tag{4}$$

Although sessions are anonymous, learning their items and behaviors from both structural and temporal perspectives can still capture the unique preferences of each session. In order to achieve personalized recommendation, for each session, we construct adjacency matrices $\mathbf{A}_{in}, \mathbf{A}_{out} \in \mathbb{R}^{n\times n}$ that separately only include in degree and out degree for the macro item sequence in chronological order, where n is the length of macro item session. Meanwhile, we aggregate item embedding with behavioral features and propagate information through these two adjacency matrices to capture the preference features of the session:

$$\mathbf{P} = \mathbf{E} \oplus \mathbf{Z}, \tag{5}$$

where $\mathbf{P} \in \mathbb{R}^{n\times 2d}$ is the representation that aggregates the behavioral temporal features of macro items with the item representation and $\mathbf{E}$ is the embedding of all macro items in the session sampled from $\mathbf{E}_g^{(L_g)}$. $\oplus$ is the concatenation operation. Then, we propagate the aggregated information into two adjacency matrices, $\mathbf{A}_{in}$ and $\mathbf{A}_{out}$, respectively, to obtain higher-order fused representations for the two views:

$$\begin{aligned} \mathbf{I}_{in} &= \mathbf{A}_{in}\mathbf{W}_{in}\mathbf{P} + \mathbf{b}_{in}, \\ \mathbf{I}_{out} &= \mathbf{A}_{out}\mathbf{W}_{out}\mathbf{P} + \mathbf{b}_{out}, \end{aligned} \tag{6}$$

where $\mathbf{W}_{in}, \mathbf{W}_{out} \in \mathbb{R}^{2d\times d}$ and $\mathbf{b}_{in}, \mathbf{b}_{out} \in \mathbb{R}^{n\times d}$ are learnable parameters. After local neighborhood propagation, we extract the intention representation of items hidden by users in their behavior. However, the extracted information may contain noise. To improve model performance and generalization ability, we introduce a gating mechanism that aggregates the original item representations through the selection of information by the reset gate and update gate with the learned intent representations. The formula is as follows:

$$\begin{aligned} \mathbf{I} &= \mathbf{I}_{in} \oplus \mathbf{I}_{out}, \\ \mathbf{U} &= \sigma(\mathbf{W}_{zi}\mathbf{I} + \mathbf{W}_{zh}\mathbf{E}), \\ \mathbf{R} &= \sigma(\mathbf{W}_{ri}\mathbf{I} + \mathbf{W}_{rh}\mathbf{E}), \\ \mathbf{E}' &= \tanh(\mathbf{W}_{ni}\mathbf{I} + \mathbf{W}_{nh}(\mathbf{R} \otimes \mathbf{E})), \\ \mathbf{X} &= (\mathbf{1} - \mathbf{U}) \otimes \mathbf{E} + \mathbf{U} \otimes \mathbf{E}', \end{aligned} \tag{7}$$

where $\mathbf{I} \in \mathbb{R}^{n\times 2d}$ represents the fusion intention representation that combines two views. $\mathbf{W}_{zi}, \mathbf{W}_{ri}, \mathbf{W}_{ni} \in \mathbb{R}^{2d\times d}$ and $\mathbf{W}_{zh}, \mathbf{W}_{rh}, \mathbf{W}_{nh} \in \mathbb{R}^{d\times d}$ are trainable parameters. $\sigma(\cdot)$ means *sigmoid* function and $\otimes$ denotes the element level multiplication operator. $\mathbf{U}$ denotes the update gate, which decides how much information from the original item embeddings to pass through. $\mathbf{R}$ represents the reset gate, which determines how much information from the original item embeddings needs to be forgotten. By using the gating mechanism to filter and update item embeddings and behavioral intention information, we have obtained the final macro item intention representation $\mathbf{X} \in \mathbb{R}^{n \times d}$ in the session.

3) Session Intention Representation Generation: In the previous section, we obtain the behavior-aware semantic representation for each macro item in the session, and the intention of the entire session can be learned based on them. In general, the user is always driven by one type of intention in a single session. However, the intentions of items within a session may be diverse, so there is inevitably noise. In order to capture the intention representation of the session more accurately, we utilize attention gating mechanism to determine the weight of each item in the session.

First, we construct the intention prototype of the session. To aggregate all item information in the session, we adopt an average pooling operation:

$$\mathbf{e}_p = \frac{1}{n} \sum_{i=1}^{n} \mathbf{e}_i, \tag{8}$$

where $\mathbf{e}_p \in \mathbb{R}^d$ is the intention prototype of the session. Then we design a gated attention layer to transfer the session intent prototype into the item representation as follows:

$$\begin{aligned} \alpha_i &= \frac{(\mathbf{W}_q \mathbf{x}_i)^T \mathbf{W}_k \mathbf{e}_p}{\sqrt{d}}, \\ \hat{\mathbf{x}}_i &= (\mathbf{1} - \alpha_i)\mathbf{x}_i + \alpha_i \mathbf{e}_p, \end{aligned} \tag{9}$$

where α_i is the attention weight of macro item v_i in the session and $\mathbf{W}_q, \mathbf{W}_k \in \mathbb{R}^{d \times d}$ are learnable parameters. In this way, we obtain the final fused item representation. Due to the attention method and GRU's information transmission mechanism, the last item in the session contains information from all previous items. Therefore, we use the representation of the last item v_n as the overall intention representation of the session:

$$\mathbf{e}_s^I = \hat{\mathbf{x}}_n, \tag{10}$$

where $\mathbf{e}_s^I \in \mathbb{R}^d$ denotes the intention representation of the session, which integrates the sequential information of items in the session, global-level neighborhood relationships, and fine-grained behavioral operations within the items.

C. Behavior Habits Learning

The purpose of this module is to capture the complete behavioral process of each item in a session to model user-level behavioral habits and guide item representation learning based on this information to achieve enhancement of user intention. In section IV-B, although we use GRU to learn the behavioral intentions corresponding to macro items, the information learned by this module is still limited to the sequential patterns in the session and cannot capture the user's whole behavior habits. For example, a user first clicks, browses, and adds items v_1 to the shopping cart, and then discovers items v_2 and v_3 that he may be interested in and browses them separately, and then he makes a purchase of v_1. In that way, the macro item sequence corresponding to this user is $[v_1, v_2, v_3, v_1]$. Our intention learning module can only learn v_1 at two time positions separately, and cannot utilize a complete behavior habit for v_1, thus unable to capture the user's potential behavioral preferences.

Since hypergraphs allow a hyperedge to connect more than two vertices, they can represent more complex relationships compared to general graph structures. In the context of behavior habits, a single item may correspond to multiple types of behaviors, making the hypergraph structure highly suitable for capturing behavior habits. Therefore, we construct a behavior habits hypergraph for each session to capture all item behavior habits within it. For the constructed hypergraph $\mathbf{G}^h = \{\mathcal{V}^h, \mathcal{E}^h\}$, $\mathcal{V}^h$ represents the set of behaviors in the behavior space and $\mathcal{E}^h$ represents the macro item set in the session. The elements in the correlation matrix $\mathcal{Q}$ corresponding to the hypergraph are defined as follows:

$$q_{ij} = |b_j^i| + \beta, \tag{11}$$

where $|b_j^i|$ means the number of behavior b_j performed on item v_i in the session and β is residual to prevent gradient explosion. Different users generate behavior habits hypergraphs that are also not identical, hence personal user behavior preferences can be extracted from them.

After obtaining the behavior habits hypergraph of the session, we utilize the hypergraph graph convolution method for message passing to integrate the behavior habits into the item representation. Inspired by [43], we design our hypergraph learning network as follows:

$$\mathbf{E}_h = \mathbf{D}_v^{-1} \mathcal{Q} \mathbf{D}_e^{-1} \mathcal{Q}^T \hat{\mathbf{E}}, \tag{12}$$

where $\hat{\mathbf{E}} \in \mathbb{R}^{n \times d}$ represents the initial item embedding in the session. $\mathbf{D}_e$ and $\mathbf{D}_v$ are degree matrices normalized in vertex and edge dimensions, respectively. Through hypergraph learning network, we achieve information propagation from nodes to hyperedges, and from hyperedges back to nodes, thereby refining the representation of items. After processing with hypergraph convolution, we obtain the item embedding representation $\mathbf{E}_h$ that integrates behavior habits information. Ultimately, we adopt the average pooling method to calculate the session representation:

$$\mathbf{e}_s^B = \frac{1}{n} \sum_{i=1}^{n} \mathbf{e}_i^h, \tag{13}$$

where $\mathbf{e}_s^B \in \mathbb{R}^d$ means the final session representation fused after integrating behavior habits into items.

D. Mutual Information Fusion

In our framework, we encode the session information from two perspectives, the session intention and the behavior habits. To generate the final session representation, we design a gated attention layer to achieve this mutual information fusion with trade-offs. The specific implementation process is as follows:

$$\begin{aligned} \gamma &= \sigma(\mathbf{W}_s(\mathbf{e}_s^I \oplus \mathbf{e}_s^B) + \mathbf{b}_s), \\ \mathbf{e}_s &= \gamma \otimes \mathbf{e}_s^I + (1 - \gamma) \otimes \mathbf{e}_s^B, \end{aligned} \tag{14}$$

where $\mathbf{W}_s \in \mathbb{R}^{2d \times d}$ and $\mathbf{b}_s \in \mathbb{R}^d$ are both learnable parameters and $\mathbf{e}_s$ is the final session representation. In this way, we successfully integrate the mutual information between the user's intention and behavior habits in the session, and we can utilize this feature vector for recommendation prediction.

E. Model Optimization and Prediction

To make the prediction process more stable, we perform L2 regularization on the session representation and item embedding separately:

$$\begin{aligned}\hat{\mathbf{e}}_s &= L_2Norm(\mathbf{e}_s),\\ \hat{\mathbf{E}}_g &= L_2Norm(\mathbf{E}_g),\end{aligned} \tag{15}$$

where $\mathbf{E}_g \in \mathbb{R}^{N\times d}$ is the initial global item embedding. Subsequently, we calculate the final prediction results and adopt cross-entropy loss as the optimizer for model training:

$$\begin{aligned}\hat{y}_i &= softmax(\hat{\mathbf{e}}_s^T \hat{\mathbf{E}}_g),\\ \mathcal{L} &= -\sum_{i=1}^{N} y_i log(\hat{y}_i),\end{aligned} \tag{16}$$

where $\hat{y}_i, y_i$ means the probability and the ground truth of item v_i being predicted, respectively.

V. Experiments

In this section, we conduct extensive experiments and analyze the experimental results in detail to answer the following questions:

- **RQ1:** Can the proposed multi-behavior session-based recommendation model BHSBR outperform other baselines?
- **RQ2:** Does each module of the BHSBR model have a positive impact on performance?
- **RQ3:** What is the effect of key parameters to the BHSBR model performance?
- **RQ4:** Does the BHSBR model capture users' intentions accurately?

We first introduce the experimental settings, including the datasets, baselines, evaluation metrics, and parameter settings. Subsequently, we analyze each experimental result to answer the questions asked above.

A. Experimental Settings

1) Datasets: We conduct extensive experiments on three real-world datasets: *Appliances*, *Computers*, and *Trivago*, which contain rich session behavior interaction data. Among them, *Appliances* and *Computers* datasets are from China's huge e-commerce platform JD.com [1], while the *Trivago* dataset is provided by the global hotel search platform trivago in the RecSys Challenge 2019 [2]:

- **Appliances:** This dataset contains session interaction records for "Appliances". It has 10 different types of behaviors, such as "add to cart", "purchase", and so on.
- **Computers:** This dataset, along with *Appliances*, belongs to JD's dataset and contains historical interaction data about "Computers". Similarly, it also includes 10 different types of behaviors.
- **Trivago:** This dataset contains interaction data about hotels, with a total of six types of operation objects being the behavior types of items. During the experiment, we only use the training set of the original dataset to repartition and conduct the experiment.

[1]https://tinyurl.com/ybo8z4yz
[2]http://www.recsyschallenge.com/2019/

TABLE I
DATASETS

Dataset Name	Appliances	Computers	Trivago
# behavior types	10	10	6
# items	77,258	96,290	183,561
# train sessions	1,294,157	416,664	260,877
# validation sessions	184,399	59,518	37,027
# test sessions	358,795	118,434	74,770

During the dataset processing phase, based on the number of sessions, we divide the training set, validation set, and testing set in the proportions of 70%, 10%, and 20% respectively. For the *Trivago* dataset, we filter out items with fewer than 5 occurrences and sessions with only one interactive item. Unlike the processing of the *Trivago* dataset, for the other two datasets, we filter out items that appeared less than 50 times. The relevant information of these datasets is shown in Table I.

2) Baselines: We compare our **BHSBR** with various recommendation model baselines to verify the effectiveness and superiority of our model.

General Session-based Recommendation Methods (SBRs):

- **GRU4Rec** [6]: It introduced GRU for the first time in session recommendation to learn user preferences.
- **SASRec** [44]: It leveraged attention mechanism to model user historical behavior and extracted session preferences.
- **STAMP** [21]: It designed a long-term and short-term memory network to capture users' long-term memory and recent interests separately.
- **SRGNN** [23]: It built a item level interaction directed graph for each session to capture the complex transformation relationships of items.
- **SGNN-HN** [24]: It utilized star graph neural networks to capture complex transformation relationships in items.
- **COTREC** [26]: It designed a self-supervised contrastive learning paradigm to achieve item-level information enhancement.
- **MCLRec** [27]: It utilized meta-learning to guide parameter updates of model enhancers, in order to improve contrast quality without increasing input data volume.

Hypergraph-enhanced SBRs:

- **DHCN** [28]: It modeled each session as a hyperedge of a hypergraph, with items as vertices, to capture high-order interactions between items.
- **HIDE** [29]: It constructed a hypergraph for each session and decoupled the click intention of the item from both macro and micro perspectives.

Multi-behavior SBRs:

- **MBHT** [13]: It designed a multi-scale transformer to capture sequential patterns and combined it with hypergraph networks to learn behavioral dependencies.
- **EMBSR** [14]: It adopted an extended self-attention network to utilize pairwise relational patterns of micro behavior to capture fine-grained user preferences.
- **PBAT** [42]: It captured user features through dynamic representation encoding and personalized pattern learn-

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON THREE DATASETS

Metric	General Session-based Recommendation							Hypergraph-based		Multi-behavior Methods			Ours	Impv.
	GRU4Rec	SASRec	STAMP	SRGNN	SGNN-HN	COTREC	MCLRec	DHCN	HIDE	MBHT	EMBSR	PBAT	BHSBR	
Appliances														
Recall@10	25.25	27.56	25.58	31.09	31.53	30.25	24.93	31.08	29.20	0.46	34.40	27.28	**36.82**	7.03%
MRR@10	13.15	14.61	14.29	16.71	16.75	10.34	11.46	12.45	11.10	0.15	17.21	10.53	**18.53**	7.67%
NDCG@10	16.01	17.67	16.97	20.11	20.24	15.00	14.63	16.82	15.37	0.23	21.26	14.42	**22.82**	7.34%
Recall@20	31.48	33.83	30.53	38.19	38.70	40.07	32.47	40.77	37.95	0.68	42.75	38.25	**45.90**	7.37%
MRR@20	13.59	15.03	14.63	17.20	17.24	11.05	11.98	13.12	11.71	0.16	17.79	11.29	**19.13**	7.53%
NDCG@20	17.58	19.22	18.22	21.90	22.05	17.58	16.52	19.26	17.58	0.28	23.38	17.19	**25.06**	7.19%
Computers														
Recall@10	5.42	7.82	7.59	11.86	12.53	16.57	13.11	17.12	10.27	2.24	15.05	3.76	**18.31**	6.95%
MRR@10	2.49	4.25	4.72	5.43	5.79	5.90	4.88	6.16	3.79	0.84	5.68	1.13	**7.69**	24.83%
NDCG@10	3.17	5.09	5.40	6.93	7.37	8.64	6.82	8.76	5.30	1.16	7.83	1.73	**10.18**	16.21%
Recall@20	7.74	9.76	8.85	16.20	16.76	23.79	17.91	23.85	14.68	3.61	20.64	6.55	**25.06**	5.07%
MRR@20	2.63	4.38	4.80	5.73	6.08	6.39	5.21	6.68	4.10	0.93	6.06	1.32	**8.15**	22.01%
NDCG@20	3.70	5.56	5.72	8.03	8.43	10.45	8.03	10.67	6.42	1.51	9.22	2.43	**11.86**	11.15%
Trivago														
Recall@10	15.33	17.53	17.65	11.90	9.96	21.26	20.87	23.56	21.93	12.50	22.78	6.51	**25.38**	7.72%
MRR@10	7.52	8.38	9.26	5.60	4.92	5.75	7.75	8.97	7.78	5.38	9.93	1.42	**11.04**	11.17%
NDCG@10	9.34	10.54	11.26	7.08	6.10	9.35	10.82	12.50	11.08	7.05	12.93	3.14	**14.15**	9.43%
Recall@20	19.90	22.53	21.21	15.77	12.94	31.82	28.55	33.56	30.86	17.27	31.10	13.99	**34.27**	2.12%
MRR@20	7.81	8.72	9.51	5.87	5.13	6.47	8.28	9.67	8.40	5.71	10.51	1.88	**11.56**	9.99%
NDCG@20	10.46	11.81	12.16	8.06	6.86	12.02	12.76	15.05	13.34	8.25	15.04	5.00	**16.39**	8.90%

ing, and designed a behavior-aware collaborative extractor to extract the semantics of behavior transitions.

3) Evaluation Metrics: To more comprehensively evaluate the performance of our model, we employ several commonly used evaluation metrics in the session-based recommendation domain to assess all models. In order to make the experiment more convincing, we select three evaluation metrics: top-K Recall (Recall@K), top-K Mean Reciprocal Rank (MRR@K), and top-K Normalized Discounted Cumulative Gain (NDCG@K). We set the value of K to [10, 20].

4) Parameter Settings: For all baseline methods, we follow the relevant configurations described in their corresponding papers, preprocess the data, and set parameters to ensure they achieve optimal results. In order to ensure the fairness and effectiveness of the experiment, we unify the prediction goal as the next item of the session without considering whether it is the target behavior. The parameter values used in our framework are set as follows. We configure our model with a batch size of 128 and set the training epochs to 30. For the *Trivago* dataset, we establish a maximum sequence length of 50, while the other two datasets are 20. Additionally, We set the embedding dimension value to 100.

B. Overall Comparison (RQ1)

We conduct performance evaluations on all baselines and our **BHSBR** in the task for predicting the next item. The evaluation results of the three datasets are shown in Table II. From the research results, we have the following summary:

- Our proposed **BHSBR** achieves the best performance on three datasets compared to all baseline methods. On average, it improved by 8.21% on *Trivago*, 14.37% on *Computers*, and 7.36% on *Appliances*, verifying the superiority of our model.
- Among three datasets, the improvement on the *Computers* dataset is more significant. A possible reason is that a significant number of behavior habits in the *Computers* dataset are "search", which means that there may be a multitude of items in each session that have only "search" interactions. Consequently, this results in a substantial amount of noise. Compared to other methods, our proposed model exhibits greater robustness and is capable of effectively mitigating the impact of noise.
- GNN-based methods generally perform better than RNN-based methods, indicating that GNN plays a significant role in modeling session structures. Among them, DHCN performs outstandingly on three datasets, proving the superiority of hypergraphs in capturing high-order information.

C. Ablation Study (RQ2)

In order to verify the rationality and effectiveness of each module design of our **BHSBR**, we conduct sufficient ablation experiments. Specifically, we set the following variants:

- **BHSBR_NH** removes the hypergraph-based behavior habit learning layer and only encodes the session representation through the behavior-aware intention learning layer.
- **BHSBR_NI** removes the entire intention learning module and only retains the behavior habit learning layer, capturing only all behavior habits in the session to encode the final representation.
- **BHSBR_NG** removes the global semantic capture process in the intent learning module, only aggregating behavioral information under macro items without utilizing global semantic information for intent enhancement.

The results of the ablation experiment are shown in Table III. From the experimental results, we can draw the following conclusions:

- Among the three datasets, **BHSBR_NH** generally performs the worst, especially in the MRR and NDCG metrics, indicating that capturing user behavior habits plays a profoundly significant role in enhancing user intention and reducing noise issues in sessions.
- The performance degradation of **BHSBR_NI** is also significant, demonstrating that there is a deep correlation between user intent and the temporal relationship between items in session recommendation. Consequently, integrating behavioral information and semantic information into the

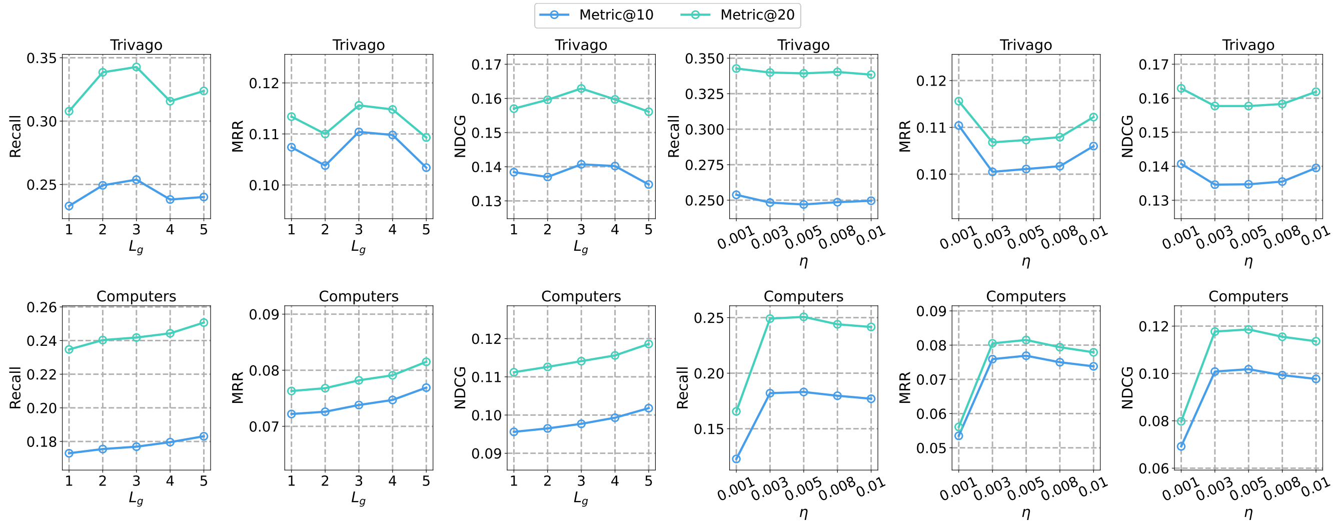

Fig. 4. The impact of hyperparameters L_g and η on *Trivago* and *Computers* datasets.

TABLE III
ABLATION STUDY ON THREE DATASETS

Dataset	Metric	BHSBR_NH	BHSBR_NI	BHSBR_NG	BHSBR
Appliances	Recall@10	34.74	35.10	36.73	**36.82**
	MRR@10	16.98	17.53	18.19	**18.53**
	NDCG@10	21.08	21.67	22.56	**22.82**
	Recall@20	43.52	44.24	45.46	**45.90**
	MRR@20	17.57	18.16	18.81	**19.13**
	NDCG@20	23.24	23.98	24.83	**25.06**
Computers	Recall@10	16.76	15.20	17.12	**18.31**
	MRR@10	6.45	6.95	7.23	**7.69**
	NDCG@10	8.82	8.88	9.42	**10.18**
	Recall@20	23.16	20.40	23.01	**25.07**
	MRR@20	6.87	7.30	7.62	**8.15**
	NDCG@20	10.44	10.18	10.84	**11.86**
Trivago	Recall@10	23.29	23.10	23.98	**25.38**
	MRR@10	9.34	10.63	10.23	**11.04**
	NDCG@10	12.59	13.53	13.38	**14.07**
	Recall@20	32.17	30.56	32.38	**34.27**
	MRR@20	9.95	11.13	10.81	**11.56**
	NDCG@20	14.83	15.39	15.52	**16.29**

session temporal relationship can better capture the user's true intention.

- The performance of **BHSBR_NG** has also decreased on three datasets. Notably, the decrease is particularly significant on the *Trivago* dataset. One possible reason is that in the *Trivago* dataset, most of the predicted items in each session have not appeared in previous interactions. Thus, after removing the global semantic capture layer, the model tends to recommend items that have already been interacted with by users, while lacking the ability to discover new items.

D. The Impact of Hyperparameters (RQ3)

To investigate the impact of different hyperparameter values on model performance, we conduct sufficient hyperparameter experiments on the layer number of global semantic capturing process L_g and the learning rate η of Adam loss optimizer on the *Trivago* and *Computers* datasets. The experimental results are shown in Fig. 4.

- We respectively set the value of L_g to $\{1, 2, 3, 4, 5\}$. From Fig. 4, it can be observed that as L_g increases, each item can aggregate richer global neighborhood information, which better captures user intent during the intention encoding process. Consequently, the model's performance improves with the elevation of L_g. However, in *Trivago* dataset, when L_g exceeds 3, the model performance actually decreases. This is because too many iterations of global graph convolution can cause items to aggregate too much neighborhood information, resulting in over-smoothing issues, and leading to a decrease in model performance. In summary, the optimal L_g value varies for different datasets, and appropriate sizes should be selected for each dataset to achieve optimal model performance.
- The values of η is examined in $\{0.001, 0.003, 0.005, 0.008, 0.01\}$. Generally speaking, the learning rate η of the optimizer controls the step size of model optimization. Setting the learning rate too low can easily slow down the convergence speed of the model and drag down its efficiency. If the learning rate is set too high, however, it may lead to missing the optimal solution. Therefore, it is necessary to make a trade-off and set an appropriate learning rate size to achieve relative optimization. From Fig. 4, we can observe that the optimal η value in the *Trivago* dataset is 0.001, while in the *Computers* dataset it is 0.005. Therefore, the optimal η values corresponding to different datasets also vary.

E. Case Study (RQ4)

To verify that our BHSBR can accurately capture user intentions, we conduct a case study and randomly sample a session 435 from the *Computers* dataset. We analyze its behavior habits hypergraph structure and semantic neighbors in detail and present its prediction results, as shown in Fig. 5. From the graph, we can observe that by capturing the behavior habits of this session, the user has engaged in a lot of behavior towards items 4 and 1963, which means that the user's main intention should be focused on these two items. Meanwhile, our designed intention learning module mines semantic information of items from a global perspective, and it

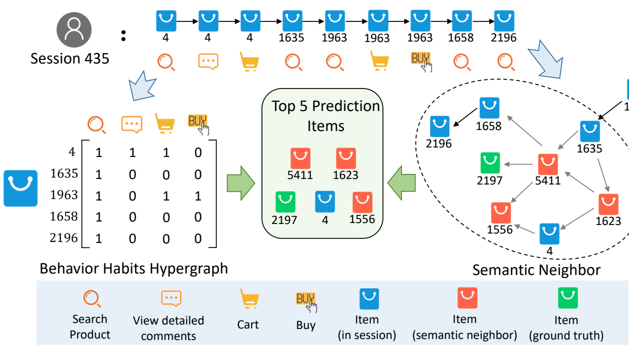

Fig. 5. A case for BHSBR Prediction on *Computers* dataset.

can be observed that in this session, except for item 1963, the other four items can be connected through semantically related items. By combining these two pieces of information, our BHSBR is able to capture that the user's intention should be associated with item 4. We provide the top-5 prediction results of this session, where we successfully predict target item 2197, and all predicted items have strong semantic connections with item 4, which proves that BHSBR can accurately capture user intentions and provide new item recommendations for users.

Acknowledgments

This work is supported in part by the National Nature Science Foundation of China under Grant Number 62102029, and in part by the Beijing Institute of Technology Research Fund Program for Young Scholars.

VI. Conclusion

In this work, we conduct data analysis on multi-behavior scenarios in recommendation models to explore how to use behavioral information to learn behavioral dependencies and better capture user intentions, and explore task settings in multi-behavior scenarios. We propose a new session-based recommendation framework BHSBR. Specifically, we adopt global GNN and GRU to learn semantic relationships and behavioral information of items in a session, and fuse them using gating mechanisms to obtain intent representations. Meanwhile, we design a hypergraph for each session to capture user-level behavior habits and achieve intention enhancement through HGNN. Extensive experiments demonstrate our model's superiority over SOTA approaches, with ablation studies confirming the efficacy of each module.

References

[1] S. Qiao, W. Zhou, J. Wen, H. Zhang, and M. Gao, "Bi-channel multiple sparse graph attention networks for session-based recommendation," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, ser. CIKM '23. Association for Computing Machinery, 2023, p. 2075–2084.

[2] Q. Chen, J. Li, Z. Guo, G. Li, and Z. Deng, "Attribute-enhanced dual channel representation learning for session-based recommendation," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, ser. CIKM '23. Association for Computing Machinery, 2023, p. 3793–3797.

[3] D. Yu, Q. Li, H. Yin, and G. Xu, "Causality-guided graph learning for session-based recommendation," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, ser. CIKM '23. Association for Computing Machinery, 2023, p. 3083–3093.

[4] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," ser. WWW '10, 2010, p. 811–820.

[5] W. Gu, S. Dong, and Z. Zeng, "Increasing recommended effectiveness with markov chains and purchase intervals," *Neural Comput. Appl.*, vol. 25, no. 5, p. 1153–1162, oct 2014.

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

[7] J. Li, P. Ren, Z. Chen, Z. Ren, and J. Ma, "Neural attentive session-based recommendation," 2017.

[8] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," 2016.

[9] S. Lai, E. Meng, F. Zhang, C. Li, B. Wang, and A. Sun, "An attribute-driven mirror graph network for session-based recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '22. Association for Computing Machinery, 2022, p. 1674–1683.

[10] D. Jin, L. Wang, Y. Zheng, G. Song, F. Jiang, X. Li, W. Lin, and S. Pan, "Dual intent enhanced graph neural network for session-based new item recommendation," 2023.

[11] J. Su, C. Chen, W. Liu, F. Wu, X. Zheng, and H. Lyu, "Enhancing hierarchy-aware graph networks with deep dual clustering for session-based recommendation," in *Proceedings of the ACM Web Conference 2023*, ser. WWW '23, 2023, p. 165–176.

[12] L. Xia, C. Huang, Y. Xu, and J. Pei, "Multi-behavior sequential recommendation with temporal graph transformer," *IEEE Transactions on Knowledge and Data Engineering*, p. 1–1, 2022.

[13] Y. Yang, C. Huang, L. Xia, Y. Liang, Y. Yu, and C. Li, "Multi-behavior hypergraph-enhanced transformer for sequential recommendation," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '22. ACM, Aug. 2022.

[14] J. Yuan, W. Ji, D.-F. Zhang, J. Pan, and X. Wang, "Micro-behavior encoding for session-based recommendation," *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2886–2899, 2022.

[15] L. Xia, C. Huang, Y. Xu, P. Dai, B. Zhang, and L. Bo, "Multiplex behavioral relation learning for recommendation via memory augmented transformer network," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. ACM, Jul. 2020.

[16] L. Guo, L. Hua, R. Jia, B. Zhao, X. Wang, and B. Cui, "Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior," ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1984–1992.

[17] E. Yuan, W. Guo, Z. He, H. Guo, C. Liu, and R. Tang, "Multi-behavior sequential transformer recommender," ser. SIGIR '22. Association for Computing Machinery, 2022, p. 1642–1652.

[18] A. Li, Z. Cheng, F. Liu, Z. Gao, W. Guan, and Y. Peng, "Disentangled graph neural networks for session-based recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 7870–7882, 2023.

[19] W. Wang, W. Zhang, S. Liu, Q. Liu, B. Zhang, L. Lin, and H. Zha, "Incorporating link prediction into multi-relational item graph modeling for session-based recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 2683–2696, 2023.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[21] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "Stamp: Short-term attention/memory priority model for session-based recommendation," ser. KDD '18, 2018, p. 1831–1839.

[22] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," 2019.

[23] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, p. 346–353, Jul. 2019.

[24] Z. Pan, F. Cai, W. Chen, H. Chen, and M. de Rijke, "Star graph neural networks for session-based recommendation," ser. CIKM '20, 2020, p. 1195–1204.

[25] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, and M. Qiu, "Global context enhanced graph neural networks for session-based recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. ACM, Jul. 2020.

[26] X. Xia, H. Yin, J. Yu, Y. Shao, and L. Cui, "Self-supervised graph co-training for session-based recommendation," 2021.

[27] X. Qin, H. Yuan, P. Zhao, J. Fang, F. Zhuang, G. Liu, Y. Liu, and V. Sheng, "Meta-optimized contrastive learning for sequential recommendation," ser. SIGIR '23, 2023, p. 89–98.

[28] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," 2022.

[29] Y. Li, C. Gao, H. Luo, D. Jin, and Y. Li, "Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation," ser. SIGIR '22, 2022, p. 1997–2002.

[30] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," ser. KDD '08, 2008, p. 650–658.

[31] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using bayesian personalized ranking for social network data," ser. WSDM '12, 2012, p. 173–182.

[32] C. Chen, W. Ma, Z. Min, Z. Wang, X. He, C. Wang, Y. Liu, and S. Ma, "Graph heterogeneous multi-relational recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 3958–3966, 05 2021.

[33] B. Jin, C. Gao, X. He, D. Jin, and Y. Li, "Multi-behavior recommendation with graph convolutional networks," ser. SIGIR '20, 2020, p. 659–668.

[34] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi, "Improving user topic interest profiles by behavior factorization," ser. WWW '15, 2015, p. 1406–1416.

[35] W. Zhang, J. Mao, Y. Cao, and C. Xu, "Multiplex graph neural networks for multi-behavior recommendation," ser. CIKM '20, 2020, p. 2313–2316.

[36] L. Xia, Y. Xu, C. Huang, P. Dai, and L. Bo, "Graph meta network for multi-behavior recommendation," ser. SIGIR '21, 2021, p. 757–766.

[37] C. Meng, H. Zhang, W. Guo, H. Guo, H. Liu, Y. Zhang, H. Zheng, R. Tang, X. Li, and R. Zhang, "Hierarchical projection enhanced multi-behavior recommendation," ser. KDD '23, 2023, p. 4649–4660.

[38] W. Wei, C. Huang, L. Xia, Y. Xu, J. Zhao, and D. Yin, "Contrastive meta learning with behavior multiplicity for recommendation," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, ser. WSDM '22. ACM, Feb. 2022.

[39] J. Xu, C. Wang, C. Wu, Y. Song, K. Zheng, X. Wang, C. Wang, G. Zhou, and K. Gai, "Multi-behavior self-supervised learning for recommendation," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '23. ACM, Jul. 2023.

[40] M. Zhou, Z. Ding, J. Tang, and D. Yin, "Micro behaviors: A new perspective in e-commerce recommender systems," *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018.

[41] W. Meng, D. Yang, and Y. Xiao, "Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. ACM, Jul. 2020.

[42] J. Su, C. Chen, Z. Lin, X. Li, W. Liu, and X. Zheng, "Personalized behavior-aware transformer for multi-behavior sequential recommendation," 2024.

[43] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," ser. AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[44] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," 2018.