Modeling Cross-session Information with Multi-interest Graph Neural Networks for the Next-item Recommendation

TING-YUN WANG, Institute of Information Management, National Yang Ming Chiao Tung University, Taiwan
CHIAO-TING CHEN, Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan
JU-CHUN HUANG, Department of Information Management and Finance, National Yang Ming Chiao Tung University, Taiwan
SZU-HAO HUANG, Department of Information Management and Finance, National Yang Ming Chiao Tung University, Taiwan

Next-item recommendation involves predicting the next item of interest of a given user from their past behavior. Users tend to browse and purchase various items on e-commerce websites according to their varied interests and needs, as reflected in their purchasing history. Most existing next-item recommendation methods aim to extract the main point of interest in each browsing session and encapsulate it in a single representation. However, past behavior sequences reflect the multiple interests of a single user, which cannot be captured by methods that focus on single-interest contexts. Indeed, multiple interests cannot be captured in a single representation, and doing so results in missing information. Therefore, we propose a model with a multi-interest structure for capturing the various interests of users from their behavior sequence. Moreover, we adopted a method based on a graph neural network to construct interest graphs based on the historical and current behavior sequences of users. These graphs can capture complex item transition patterns related to different interests. In experiments, the proposed method outperforms state-of-the-art session-based recommendation systems on three real-world data sets, achieving 4% improvement of Recall over the SOTAs on Jdata dataset.

CCS Concepts: • **Information systems → Recommender systems**; • **Computing methodologies → Machine learning algorithms**.

Additional Key Words and Phrases: Next-Item Recommendation, Multi-Interest, Graph Neural Network

1 INTRODUCTION

Recommender systems (RSs) help users filter out irrelevant information and select the appropriate resources; RSs have been used in various applications, such as e-commerce, streaming services, and hotel booking websites. An efficient RS helps users determine their interests and needs and helps businesses to profit. Therefore, it is essential to satisfy the unique needs of every user of an online platform. Numerous studies have developed RSs. For example, session-based RSs (SBRSs) [34] have attracted considerable attention among businesses and

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract MOST 110-2221-E-A49-101 and Contract MOST 110-2622-8-009-014-TM1; and in part bythe Financial Technology (FinTech) Innovation Research Center, National Yang Ming Chiao Tung University.
Authors' addresses: Ting-Yun Wang, tingyun199620@gmail.com, Institute of Information Management, National Yang Ming Chiao Tung University, Hsinchu, Taiwan; Chiao-Ting Chen, rsps971130.cs09@nycu.edu.tw, Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan; Ju-Chun Huang, mickey20125.ms05@nctu.edu.tw, Department of Information Management and Finance, National Yang Ming Chiao Tung University, Hsinchu, Taiwan; Szu-Hao Huang, szuhaohuang@nycu.edu.tw, Department of Information Management and Finance, National Yang Ming Chiao Tung University, Hsinchu, Taiwan.

1556-4681/2022/5-ART $15.00
https://doi.org/10.1145/3532192

academia. Users' browsing records directly indicate their interests and preferences; therefore, RSs should make recommendations according to these records. Unlike most RSs, which make recommendations according to users' data and historical records, SBRSs use short-time session data on users' behaviors to make recommendations. Thus, SBRSs completely exploit the dependencies between actions in sequential behavior data to predict the next item of interest for a given user.

Fig. 1. Example of a session containing items of various levels of interest.

Fig. 2. Example of multi-interest context within historical and current behavior sequences.

Next-item [28, 45] RSs predict and recommend the next item that a user might be interested in according to the previous behavior sequence of the user. Most next-item RSs determine a user's main interest in each session and represent it using a single session embedding [7, 11, 14, 35, 40]. However, users might browse various items in a single session because their interests and needs are varied. The multiple interests of users cannot be accurately represented by a single embedding representation. Moreover, a single representation is insufficient for reflecting the multi-interest context of behavior sequences. Thus, a single representation imprecisely reflects multiple interests, which leads to inaccurate recommendations by the RS. Consider the example in the first part of Figure 1, which displays the session browsing behavior of a user named Alice. Alice spent her leisure time shopping online and first viewed cellphones and cameras because she enjoys taking photos of her life. Subsequently, she searched for dresses and jackets. Thereafter, Alice searched for hiking boots, which she also previously searched for. She then viewed pressed powders, jumpsuits, and lipsticks. This example reflects the typical user who may view items in several categories when browsing to satisfy their various interests and needs. The second part of Figure 1 displays the categories corresponding to the searched items. In the browsing session, Alice was interested in products in the categories of clothes, makeup, electronics, and outdoor products. A suitable next-item RS should identify Alice's multiple interests from her viewing history and then generate interest-based representations. As depicted in Figure 2, the multi-interest context exists not only in a user's current browsing sequence but also in their historical browsing sequences because their interest evolves over the long term. Alice had searched for electronics and outdoor products for a long time. The current browsing sequence only indicates Alice's interest toward outdoor products on the basis of her clicking on hiking boots. However, if both Alice's current and historical browsing sequences are considered, the RS can determine that Alice was searching for outdoor products and might be interested in going mountain climbing. Therefore, next-item RSs should extract the multi-interest context from current and historical browsing sequences to generate precise interest-based representations. On the basis of these representations, a recommendation list comprising items satisfying users' different interests can be generated. By using the aforementioned method, RSs can accurately represent the diverse interests of users using the corresponding embeddings to make more appropriate recommendations.

SBRSs predict the next item according to sequences of previous behaviors. Various approaches have been proposed for completely modeling previous search-behavior patterns and making precise next-item recommendations. Early research assumed that the next item is only related to the previously searched ones according to a Markov chain (MC) [20]. Methods in current research consider the dependence among all searched items to be a crucial feature, integrate the browsing information for an entire session, and consider not only the searched items but also the item transition patterns. Behavior sequence data are similar to time-series data; therefore, recurrent neural networks (RNNs) are commonly used to process browsing session data [7]. However, RNNs regard the relations between items in sequences as rigid dependencies, which runs counter to the random and complicated nature of user behavior. Thus, RNNs are combined with the attention mechanism to capture the crucial information in a sequence [11, 14]. However, the aforementioned methods have only considered item-level transitions of rigid order and neglected more complex patterns, such as context-level transitions.

Graph neural network (GNNs) have been used in SBRSs to capture the complicated transition patterns in user behavior sequences and to model the dependencies of item sequences in order [40–42]. The aforementioned GNN-based methods generate precise item representations for forming comprehensive session embeddings to predict the next item. In this study, we posit that the graph structure is suitable for fitting session data; such a fitting is realized by treating each item in a sequence as a node and connecting consecutive nodes with edges. Moreover, GNN-based methods are highly capable of modeling the rich relations between nodes and providing an accurate representation of each node. Although the aforementioned GNN-based methods achieve excellent performance in modeling user behavior sequences, they neglect the multi-interest context in a sequence and only model the most significant interest of each user. In most GNN-based methods, each neighborhood node is considered to have the same importance as the center node, but this may not be the case. For example, if a user clicks on milk, toast, and a T-shirt in order, the milk and T-shirt nodes have a different influence on the toast node. A user is driven by a similar interest to click milk and toast because they belong to the breakfast category. However, the correlation between toast and a T-shirt is not as strong as that between toast and milk; thus, correlations should not be treated as equal. However, most GNN-based methods only consider the connections in a session graph and ignore the interest that prompts users to click on a specific item. Moreover, they tend to integrate all features into a single representation, which is insufficiently accurate. Therefore, in this study, we constructed a multi-interest graph to not only model rich item transition patterns but consider item dependencies in relation to various interests; thus, the proposed model generates superior interest-based representations and thus more accurate recommendations.

Numerous methods have been proposed for modeling the multi-interest context in sequence data. In a mixture-channel purpose routing network (MCPRN) [38], the purposes of each item are recognized and each item is assigned to a corresponding channel. MIND [10] utilizes the dynamic routing mechanism to capture a user's multiple interests from a behavior sequence. ComiRec [2] is a comprehensive framework that includes a multi-interest extraction module for extracting the multiple interests of users and an aggregation module for making highly diverse recommendations. In the aforementioned methods, each item is processed according to its corresponding interests, and multiple representations are obtained for different interests. By using a multi-interest framework, each item and browsing session can be represented by accurate representations. However, methods based on multi-interest frameworks mainly involve the use of approaches such as RNNs, self-attention mechanisms, and dynamic routing. The GNN, which performs well in session recommendation, should be used to capture complex item transition patterns related to multiple interests.

In this paper, we propose our multi-interest GNN (MI-GNN) method for capturing the user's various interests from their current and historical behavior sequences. By constructing the multiple interest graphs for current and historical sequences respectively, the user's interests are comprehensively recognized with our specially designed GNN. Furthermore, because users' interests evolve over a long time, we integrated users' overall interests from current and historical behavior sequences to generate precise context representations for each interest. To consider users' different interests related to diversified target items when making recommendations, we used a target-aware attention mechanism for obtaining the overall representation. The main contributions of this study are as follows:

- A multi-interest graph was constructed to model complex item transition patterns related to different interests, which can be used to generate precise representations of interests from item and sequence embeddings.
- Current and historical interest graphs were constructed separately to capture item dependencies in terms of users' different interests for comprehensively extracting users' diversified interests.
- The proposed method outperformed its counterparts when applied to three real-world data sets from an e-commerce website, a hotel searching website, and an online forum.

The rest of this paper is organized as follows. Section 2 describes studies related to this research. Section 3.1 defines the recommendation problem. Section 3.2 and 3.3 introduce the proposed methods for processing current and historical behavior sequences, respectively. Section 3.4 describes the method for generating representations for current and historical interests. Section 3.5 presents an explanation of how historical representations can be combined with the current representation to represent the multiple interests in the current sequence. Section 4 describes our experiments and presents the experimental results. Finally, section 5 concludes the study.

2 RELATED WORK

Because RSs are crucial to many businesses, several researchers have attempted to develop reliable RSs. This section presents a review of some techniques that have been developed for session-based recommendation. We developed our method on the basis of these techniques. Most conventional recommendation models use users' browsing history to identify suitable items for recommendation, and this serves as the basis of the proposed method. With the development of deep-learning-based methods, researchers have developed methods that involve the use of an RNN with an attention mechanism as the main model structure for capturing sequential features and filtering out irrelevant information. Several researchers have attempted to enhance the efficiency and accuracy of recommenders. For example, Nilashi et al. [15] proposed a recommender system based on collaborative filtering. Boratto et al. [1] presented an approach that involves prefiltering the items that a user has evaluated and removing the items that the user did not like. Moreover, some studies [22, 27] have discussed the evaluation of recommender systems. GNN-based recommendation methods have achieved high performance; therefore, numerous studies have used GNNs to capture complex item transition patterns. Most RSs aim to extract the main interest from a behavior sequence. However, methods based on multi-interest frameworks explore the multi-interest context in a sequence. The proposed approach involves using a GNN-based method to construct a model with the multi-interest structure.

2.1 Session-based Recommender Systems

Collaborative Filtering (CF) [25] is the main method used in general RSs; it uses data on user behavior to identify user interests to make recommendations. Matrix factorization (MF)-based methods [6, 9] aim at factorizing a user–item rating matrix into two matrices that represent the user and item representation vectors, respectively. The item k-nearest neighbors (iKNN) algorithm [23] computes the item similarity and selects the most similar items for recommendation. The drawback of the aforementioned methods is that they fail to capture the sequential context. MC-based sequential methods model the transition probabilities between items and assume that the next action is only related to previous actions. The sequential optimization algorithm uses Markov decision processes to make recommendations [26]. The factorized personalized MC method [20] is a combination of the MF and MC methods and is used to model users' general interests and the transition pattern of the user's current actions. MC-based models have served as the basis for subsequent methods, but their prediction ability is limited because they treat each action transition as having an independent pattern.

Deep learning has been widely used in RSs, and RNNs have been used for analyzing sequential data. Moreover, gated recurrent units (GRUs) and long short-term memory are two popular RNN structures based on unique memory cell structures. These structures can adjust the amount of long- and short-term memory remembered in a cell by automatically learning the model weights. Therefore, numerous researchers have used RNNs in SBRSs to model session data on resources with order or time conditions. In GRU4Rec [7], the GRU structure is used to model the user's interests in general in a session and predict the next item. The aforementioned model exhibits high performance in session-based recommendation and has influenced numerous studies. Because of the success of GRU4Rec, studies have attempted to improve it [17, 29].

Studies have used convolutional neural networks (CNNs), which are widely used in computer vision, to make session-based recommendations. Through the filtering and pooling processes of CNNs, CNN-based models can capture patterns in sequences in a manner that transcends the limitations entailed by a rigid order and can consider the local features and union-level dependencies in sequences. The Caser model [30] embeds a behavior sequence into an "image" and explores the horizontal and vertical convolutional layers to extract different levels of patterns from the sequence. NextItNet [43] improves CNN-based methods to better capture long-range dependencies from a sequence.

GNNs [24] have been widely used to solve tasks with graph-structured data. These networks can model the relations between entities in a graph and represent each entity precisely. Session data can be easily represented using a graph structure, in which nodes represent items and edges represent the transitions between consecutive items.

The SR-GNN [40] uses the GNN to perform session recommendation. It aims at modeling the complex item transition patterns in each session. The SR-GNN constructs a session graph for each session and uses a GGNN [13] to acquire the node embedding by aggregating the information from neighborhood nodes. Subsequently, the SR-GNN acquires the session embedding by concatenating the local and global embeddings to make precise recommendations.GNN-based methods can accurately capture the complex item transition patterns from a sequence and represent each item with a precise embedding. Thus, these methods outperform most recommendation methods that involve the use of RNNs and attention mechanisms. GNN-based methods extract the main interest but neglect the multi-interest context in a sequence. Thus, in the proposed method, multi-interest graphs are constructed to model the multiple interests of users.

2.2 Session-based Recommender Systems with Attention Mechanisms

Attention mechanisms [31] allow models to focus automatically on important information within a large data set by multiplying different parts of the input with different weights. The prediction accuracy of a model with an attention mechanism increases as it acquires more related information. Therefore, attention mechanism–based methods have been developed for many tasks for different domains, such as natural language processing, facial recognition, and item recommendation. NARM [11] combines a hybrid encoder with the attention mechanism to extract a user's main purpose from their behavior sequence, and this model achieves superior results for long-session data. STAMP [14] uses a short-term attention or memory priority model to capture the user's general preferences and regards the last action as the local preference. SASRec [8] utilizes the self-attention method to identify relevant items for recommendation from historical behavioral records at each time step and captures long-term semantics. ATEM [35] uses an attention-based transaction embedding model to assign different weights to each item in a transaction without assuming the order. Many methods involve the use of the attention mechanism to focus the developed model on certain actions to achieve accurate recommendations [3, 18, 33]. This technique is also adopted in the proposed method.

Most of the aforementioned methods model item-level dependencies and the sequential features of items in a rigid order. However, users' behaviors are complex, and a model must deeply analyze the context of users' sequential behaviors. Therefore, GNN-based methods are used to consider complicated item transition patterns.

Based on the success of the previous GNN-based methods, the following methods are proposed to develop different kinds of graph to recommend. The FGNN [16] utilizes a weighted graph attentional layer to aggregate the information of neighboring nodes by considering the edge weights to compute the attention scores of these nodes. The FGNN considers the different levels of influence between neighboring nodes. Since the order of an item in a sequence is neglected in the aforementioned GNN-based methods, the PA-GGAN [32] adds a positional embedding to an item embedding according to the order of the item in a sequence. Moreover, the GCE-GNN [39] constructs a session graph and global graph to not only model the item transition patterns within a session but also introduce cross-session information for enhancing the predictive ability.

Since the target item pool is too large to select the correct item from, the TA-GNN [42] considers the target items when representing a browsing session with an more precise embedding vector. The aforementioned network constructs a target-aware session embedding by using the soft attention mechanism to compute the relation scores between each item in a session and the target items. Therefore, in this network, different interests in a session are activated according to distinct target items. The TA-GNN can accurately recommend items from a large target item pool, and it achieved high performance on two commonly used session data sets.

2.3 Recommender Systems with Multi-Interest Framework

Users are driven by different interests when they click on different types of items when browsing a website. Therefore, it is insufficient to only capture the main interest in a sequence representing a diverse set of items. In this section, we describe some models that extract the multi-interest context from sequence data to predict the next item. In an MCPRN [38], a multi-channel framework is used to capture multiple purposes from a behavior sequence by identifying the purpose of each item and routing them into corresponding channels. The HLN [5] performs group learning by identifying a group of items reflecting the same preference and avoids learning repeated preferences. The aforementioned two networks are modified RNN-based networks, which outperform the RNN-based method of capturing the main interest only. MIND [10] uses the dynamic routing mechanism to develop a multi-interest extractor layer and the label-aware attention technique to learn user representations. ComiRec [2] uses two methods to construct a multi-interest extraction module: the dynamic routing and self-attentive methods. Moreover, it uses an aggregation module that makes recommendations according to the results of the multi-interest extraction module; thus, ComiRec provides accurate and diverse recommendation results.

A multi-interest framework can be used to perform next-basket recommendation. Int2Ba [37] is a modified RNN-based model that can identify, model, and respond to heterogeneous user intentions. IntNet [36] is a model with a hierarchical framework that captures the intentions that drive user actions and considers the durations of these intentions. For achieving accurate next-basket recommendation, the aforementioned models model the multi-interest context among baskets to satisfy users' multiple interests.

Most existing multi-interest frameworks have a multichannel structure and utilize methods such as RNNs, the self-attention mechanism, and dynamic routing. These frameworks outperform those that only model the main interest. However, multi-interest frameworks solely consider item-level transition patterns and model sequential data in a rigid order, which limits their performance. Therefore, to achieve superior recommendation performance, we developed a GNN-based model with an multi-interest structure to capture complex item transition patterns related to multiple interests from current and historical behavior sequences.

3 PROPOSED METHOD

3.1 Problem Statement

The aim of session recommendation is to predict the next item that a user will click on according to their current and historical item interaction sequences. The current and historical item interaction sequences provide information regarding users' preferences and are used in RSs to establish a recommendation list. In real-world scenarios, the item pool contains data on up to millions or billions of products that the recommendation system could select from; thus, accurately recommending items that users might be interested in is challenging.

Let $V = \{v_1, \ldots, v_{|V|}\}$ denote the set comprising the unique items appearing in all the sequences, and let $|V|$ denote the number of unique items in all the sequences. A user has multiple sequences denoted as $S = \{s_1, \ldots, s_{|S|}\}$, and $|S|$ represents the number of sequences for a user. Every sequence includes the interacted items in order and is represented as $s = \{v_1, \ldots, v_{|s|}\}(s \in S)$. The current sequence $s_c = \{s_n\}$ contains $t-1$ items in order, which is expressed as $s_c = \{v_1, \ldots, v_{t-1}\}$. The aim is to predict the next item v_t of the current sequence. All the sequences that occurred before s_n form the set of historical sequences, which is denoted as $s_h = \{s_1, \ldots, s_{n-1}\}$. Given both s_h and s_c, the RS generates a recommendation list in which v_t has the highest probability to have a high rank.

A sequence contains several items that reflect a user's multiple interests; therefore, attempts should be made to capture a user's various interests and use multiple representations to represent different interests. The total number of interests is set as k. Because a person's interests necessarily evolves over the long term, it is difficult to comprehensively model the user's interests using only the short-time current sequence. Therefore, in our method, the user's historical sequences are integrated with their current sequence by constructing a historical session graph and current session graph to model the complex transition patterns among items and combine representations. Through the aforementioned integration, we could not only extract a user's interests from their sequential behavior but also focus on their immediate interests reflected in the current behavioral sequence. Moreover, because we considered that the user actions within a sequence are more similar than are those from different sequences, we could capture users' interests from each historical sequence and model the relationship between their past and current interests. All the items appearing in historical sequences are represented on a large historical session graph, and the current sequence is represented on a small current session graph. In the aforementioned graphs, nodes represent items and directed edges represent the transitions between one item and the following one. Furthermore, the multiple interests expressed in the sequences are considered; this is done to convert the historical and current session graphs into k interest graphs, which are denoted as $G^h = \{g_1^h, g_2^h, g_3^h, \ldots, g_k^h\}$ and $G^c = \{g_1^c, g_2^c, g_3^c, \ldots, g_k^c\}$, respectively. The aforementioned interest graphs capture the complicated relationships between items of similar interest. Subsequently, k GGNNs are used to aggregate the information between the nodes related to k interests. For a given interest, the parameters of the GGNN are identical when propagating information in the historical and current interest graphs. Subsequently, the representations of every interest are generated from the node embedding through the attention mechanism and by integrating the current and historical information. Finally, due to the massive size of the recommendation item pool, the diversity of the target items should be considered. Therefore, the target-aware attention mechanism should be used to combine the representations of every interest. The candidate item is considered to obtain the final representation, where the candidate item has a high probability of being output. The set of final representations is used to predict the recommendation scores of all the items.

The overall structure of the proposed model is illustrated in Figure 3. We firstly construct the current and historical interest graphs. Next, multiple interest-based representations are generated by combining interest-based current and historical representations which are denoted as c1-c3 and h1-h3, respectively. Before predicting recommendation scores for each target item, we use the target-aware attention to take all target items v1-vn into accounts. Then, we can obtain s1-sn as the representations of the embeddings. Finally, the recommendation scores for all items are predicted by calculating the inner product of each target item and the context representation. For the Reddit data set, the Recall@20 and MRR@20 values of the H-RNN model were 61.8% and 33.88%, respectively, and the Recall@20 and MRR@20 values of the HierTCN model were 63.96% and 34.27%, respectively. The H-RNN and HierTCN models model a user's long- and short-term interests.

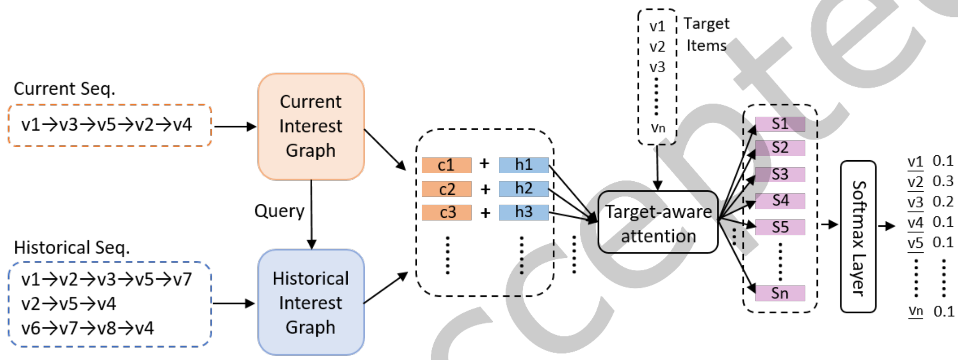

Fig. 3. Overview of proposed model that captures the multi-interest context from current (c1, c2, c3 and so on) and historical (h1, h2, h3 and so on) behavior sequences to make session-based recommendations.

3.2 Historical Sequences Pre-processing

Because human interests evolve over the long term, it is insufficient to model the short-term current sequence of user interaction. The information from current and historical sequences should be considered to comprehensively model the user's various interests.

3.2.1 Historical Session Graph. To model the overall interest in historical sequences, the corresponding historical sequence set s_h of the current sequence is used to construct a large historical cross-session graph G^h. This graph contains items from all historical sequences [point (a) to point (b) in Figure 4. $v^h = \left[v_1^h \dots v_n^h\right]$ denotes the set of unique items that are represented by nodes, and the connections between two successive items in a sequence is represented by directed edges. To form the connection matrix $\mathbf{A}_h$, the ingoing and outgoing adjacency matrices, namely $\mathbf{A}_h^i$ and $\mathbf{A}_h^o$, respectively, are concatenated. These two adjacency matrices represent the normalized incoming and outgoing edges, respectively, in the graph G^h, and can be used to model the complex information propagation between nodes. Figure 5 illustrates an example of the construction of ingoing and outgoing adjacency matrices. In the aforementioned figure, normalization is conducted for the same row. The connection matrix $\mathbf{A}_h$ is shared by the historical interest graphs and used as the input of the GGNN to obtain the node embedding.

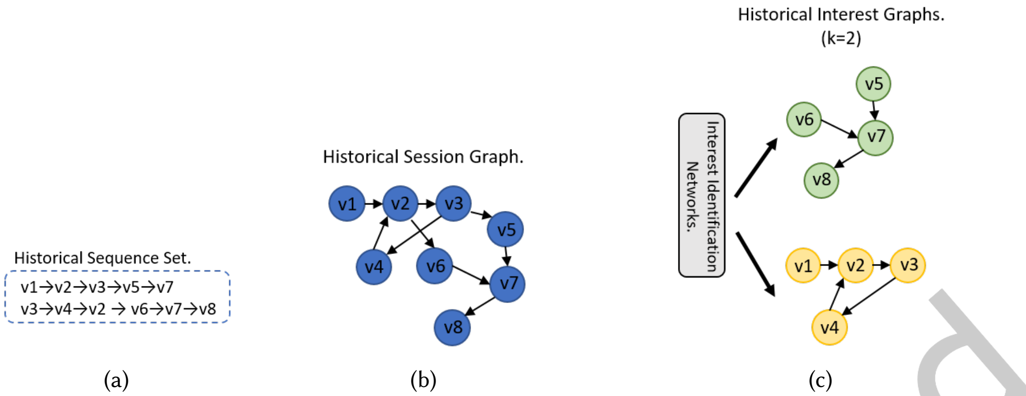

Fig. 4. Construction of historical interest graphs from historical sequences by the adopted interest identification networks.

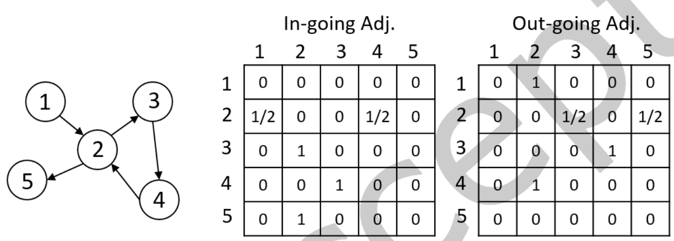

Fig. 5. Example of construction of ingoing and outgoing adjacency matrices.

3.2.2 Interest Identification Networks. Historical sequences contain various items that represent the diverse interests of users. To precisely model the item transition patterns in a graph for a specific interest, interest identification networks (IINs) are used to identify the interest that prompts a user to browse an item. Therefore, each item belongs to a specific interest graph, which prevents the propagation of unnecessary information from items belonging to different interests. First, all the items in the historical sequences are transformed into a D-dimensional embedding representation $\mathbf{v} \in \mathbf{W}_e$, where $\mathbf{W}_e \in \mathbb{R}^{D\times|V|}$ is the embedding matrix of all the items. Then, $\mathbf{W}_I \in \mathbb{R}^{D\times k}$ is used as the interest filter and the item embedding is used as the input to compute the correlation between an item and a specific interest for a D-dimensional embedding representation and k interests. The aforementioned process can be expressed as follows:

$$\mathbf{a}_{i,m} = \mathbf{v}_i^\top \mathbf{W}_I[:, m], \quad m \in \{1, \ldots, k\} \tag{1}$$

where $\mathbf{a}_{i,m}$ is the correlation between item v_i and the m^{th} interest and $\mathbf{W}_I[:, m]$ is the m^{th} column of $\mathbf{W}_I$.

Subsequently, to identify the interests that drive users to browse items, the Gumbel-softmax operation is performed using the computed correlation scores as the input:

$$\mathbf{y}_{i,m} = \frac{\exp\left(\left(\log\left(\mathbf{a}_{i,m}\right) + \pi_j\right)/\tau\right)}{\sum_{h=1}^{k} \exp\left(\left(\log\left(\mathbf{a}_{i,m}\right) + \pi_h\right)/\tau\right)}, \quad m \in \{1, \ldots, k\} \tag{2}$$

where π_j is the j^{st} Gumbel noise obtained from the Gumbel distribution with a mean of 0 and a standard deviation of 1. The hyperparameter τ denotes the temperature. When τ is close to 0^+, $[\mathbf{y}_{i,1} \ldots \mathbf{y}_{i,k}]$ is almost a one-hot vector, which indicates that each item is related to a specific interest. In our method, τ is set to 0.01.

3.2.3 Historical Interest Graph. On the basis of k interests, k historical interest graphs can be calculated, which denote as $G^h = \{g_1^h, g_2^h, g_3^h, \ldots, g_k^h\}$. In the m^{th} interest graph, each node represents a unique item v_i from all the historical sequences. Certain mathematical operations are performed to obtain the state of affairs where only the item belonging to the m^{th} interest influences the m^{th} historical interest graph. An item that does not belong to the m^{th} interest does not have any effect on and does not appear in the m^{th} historical interest graph. The aforementioned mathematical operations are described as follows. First, the interest-based item embedding of item $\mathbf{v_i}$ related to the m^{th} interest is generated as follows:

$$\mathbf{v}_{i,m} = \mathbf{y}_{i,m} * \mathbf{v}_i \tag{3}$$

Because of the hyperparameter setting of the Gumbel-softmax operation, $\mathbf{y}_{i,m}$ is a scalar close to 0 or 1 depending on whether the interest that drives the user to search the item v_i is the m^{th} interest. Therefore, if $\mathbf{y}_{i,m}$ is approximately 0, $\mathbf{v}_{i,m}$ is an approximately zero vector in the m^{th} historical interest graph. Thus, the item v_m would not appear in the m^{th} historical interest graph, and no information propagation would occur from or to this item. By contrast, if $\mathbf{y}_{i,m}$ is approximately 1, $\mathbf{v}_{i,m}$ is approximately equal to the original item embedding in the m^{th} historical interest graph. The item v_i would aggregate information from the other items appearing in the m^{th} historical interest graph, which would prevent insufficient information aggregation between items associated with different interests. Finally, the m^{th} historical interest graph g_m^h can be generated with the following initial node embedding: $\mathbf{v}_m^h = \left[\mathbf{v}_{1,m}^h \ldots \mathbf{v}_{n,m}^h\right]$. In $\mathbf{v}_m^h$, some items that do not belong to the m^{th} interest are represented by zero vectors, and no information propagation occurs from or to the nodes of these items. The aforementioned process is illustrated in Figure 4.

3.2.4 Multi-Interest Gated GNNs. To learn the node embedding vectors for various interests, k GGNNs are constructed for k interests. Each GGNN focuses on a specific interest and learns the relationships between items related to this interest. The developed GGNNs can suitably aggregate the information of neighboring nodes by using the constructed connection matrix $\mathbf{A}_h$, which considers the incoming and outgoing edges of the graph. For the m^{th} interest, the m^{th} GGNN uses $\mathbf{v}_m^h = \left[\mathbf{v}_{1,j}^h \ldots \mathbf{v}_{n,m}^h\right]$ as the initial node embedding and aggregates the information of K-hop neighbors by using K updating steps to obtain $\mathbf{v}_{i,m}^{h,K}$, which is the interest-based node embedding vector of item v_i^h for the m^{th} interest. In this study, K is set as 1. The processing of session graphs by a GGNN is based on the operations of the SR-GNN. The processing of session graphs by a GGNN can be expressed as follows:

$$\mathbf{a}_{i,m}^{h,K} = \mathbf{A}_{h,i:}\left[\mathbf{v}_{1,m}^{h,K-1}, \ldots, \mathbf{v}_{n,m}^{h,K-1}\right]^\top \mathbf{H}_m + \mathbf{b}_m \tag{4}$$

$$\mathbf{z}_{i,m}^{h,K} = \sigma\left(\mathbf{W}_{z,m}\mathbf{a}_{i,m}^{h,K} + \mathbf{U}_{z,m}\mathbf{v}_{i,m}^{h,K-1}\right) \tag{5}$$

$$\mathbf{r}_{i,m}^{h,K} = \sigma\left(\mathbf{W}_{r,m}\mathbf{a}_{i,m}^{h,K} + \mathbf{U}_{r,m}\mathbf{v}_{i,m}^{h,K-1}\right) \tag{6}$$

$$\tilde{\mathbf{v}}_{i,m}^{h,K} = \tanh\left(\mathbf{W}_{o,m}\mathbf{a}_{i,m}^{h,K} + \mathbf{U}_{o,m}\left(\mathbf{r}_{i,m}^{h,K} \odot \mathbf{v}_{i,m}^{h,K-1}\right)\right) \tag{7}$$

$$\mathbf{v}_{i,m}^{h,K} = \left(1 - \mathbf{z}_{s,i,m}^{K}\right) \odot \mathbf{v}_{i,m}^{h,K-1} + \mathbf{z}_{i,m}^{h,K} \odot \tilde{\mathbf{v}}_{i,m}^{h,K} \tag{8}$$

where $\mathbf{H} \in \mathbb{R}^{D\times 2D}$ and $\mathbf{b}_m \in \mathbb{R}^D$ are the weight and bias, respectively. Moreover, $\mathbf{z}_i$ and $\mathbf{r}_i$ are the reset and update gates, respectively, which control the information from previous and current timestamps.

After K times aggregation of GGNNs, the interest-based node embedding $\left[\mathbf{v}_{1,m}^{h,K} \ldots \mathbf{v}_{n,m}^{h,K}\right]$ of each item in the interest graph g_m^h is obtained.

3.2.5 Interest-based Historical Sequence Embedding. The embedding of each historical sequence in s_h for the m^{th} interest is obtained by aggregating the node embeddings of all the items in the sequence according to the corresponding interest identification vectors $\mathbf{y}_i$. If the sequence is highly related to the specific interest, it should have strong representations to reflect the interest.

$$\mathbf{e}_{l,m} = \sum_{i=1}^{|s_l|} \mathbf{y}_{i,m} \mathbf{v}_{i,m}^{h,K} \tag{9}$$

where s_l is the l^{th} sequence in the historical sequence set s_h and $\mathbf{e}_{l,m}$ denotes the sequence embedding of s_l for the m^{th} interest. $\mathbf{y}_{i,m}$ is the interest identification y_i for the m^{th} interest.

After obtaining the embedding of each historical sequence for every interest, these embeddings are used to generate the historical embedding of each interest, which is described later in this paper.

3.3 Current Sequence Pre-processing

The current item sequence reflects the interests of users over relatively short periods. Most users tend to browse a diverse range of items to fulfill their needs; therefore, a variety of underlying interests cause users to search for different items from a sequence. Consequently, the multiple interests of users, rather than only their main interest, should be extracted from their current sequence.

3.3.1 Current Session Graph. For modeling the complex item transition patterns in the current sequence, each sequence is transformed into a directed session graph G^c. For G^c, $v^c = \left[v_1^c, \ldots, v_n^c\right]$ denotes the items in sequence s_c, which represent nodes. The directed edges in G^c are represented by the transitions between an item and the following one. Moreover, a connection matrix $\mathbf{A}_c$ is constructed by combining the ingoing and outgoing adjacency matrices, namely $\mathbf{A}_c^i$ and $\mathbf{A}_c^o$, respectively, to account for the complex transitions between items. However, two connected items are not necessarily related because the browsing behavior of a user is sometimes random and depends on the user's various interests. Therefore, k current interest graphs are constructed to redefine the relationships between items for k interests. The construction process is described as follows.

3.3.2 Interest-based Attention Matrices. Item nodes tend to have different relationships with their neighboring nodes for different interests; therefore, the same connection matrix should not be used to propagate information between nodes in different interest graphs. Because the current sequence contains items that a user has interacted with over a short-term period, the current session graph does not contain as many item nodes as does the historical session graph. If each item in a current session graph is correlated with a specific interest (as is the case in historical interest graphs), the graph has a high probability of containing many isolated nodes. Therefore, instead of correlating each item with a single interest when constructing interest graphs, multiple-interest-based attention matrices are generated to model item transition patterns for various interests. First, an item in the current sequence is transformed to a D-dimensional embedding by using W_e as the embedding matrix. Subsequently, k self-attention processes are performed to compute the relation scores between two items for k interests as follows:

$$\mathbf{g}_{ij}^m = \left\langle \mathbf{W}_m^Q \mathbf{v}_i, \mathbf{W}_m^K \mathbf{v}_j \right\rangle \tag{10}$$

where $\mathbf{g}_{i,j}^m$ denotes the relation score between nodes i and j for the m^{th} interest and $\mathbf{W}_m^Q, \mathbf{W}_m^K \in \mathbb{R}^{d \times d}$ are the query part and key part of the projection matrices for the m^{th} interest, respectively.

Next, for considering the real edges of the graph, masked attention is performed with the ingoing and outgoing adjacency matrices, namely $\mathbf{A}_c^i$ and $\mathbf{A}_c^o$, respectively as follows:

$$\boldsymbol{\alpha}_{ij}^{m,*} = \frac{\exp\left(\mathbf{g}_{ij}^m \mathbf{A}_{c,ij}^*\right)}{\sum_{k=1}^n \exp\left(\mathbf{g}_{ik}^m \mathbf{A}_{c,ik}^*\right)} \tag{11}$$

where $*$ denotes the ingoing or outgoing relations and $\boldsymbol{\alpha}_i^{m,*} \in \mathbb{R}^{1\times n}$ represents the normalized relation score of node v_i for the m^{th} interest. By performing the aforementioned mathematical operation, the real connections in a graph and the relationships between the nodes for various interests can be considered. The $\boldsymbol{\alpha}$ values of all nodes are generated for k interests to form k interest-based attention matrices $\left[\mathbf{A}'_{c,1}, \ldots, \mathbf{A}'_{c,k}\right]$, which are used as the adjacency matrices of the corresponding current interest graphs.

3.3.3 Current Interest Graph. According to k different interests, k current interest graphs $G^c = \{g_1^c, \ldots, g_k^c\}$ are constructed. In the m^{th} interest graph, each node denotes an item v_i in the current sequence. When the interest-based attention matrix is used as the adjacency matrix, each current interest graph has different attention weights between the same item node pair. Therefore, a node can aggregate information from neighborhood nodes for different interests. If a node pair has no strong relation for a specific interest, they are designed not to propagate considerable information to each other in the interest graph, and vice versa. When adhering to the aforementioned condition, nodes of the same interest would aggregate information from each other and nodes of different interests would not propagate unnecessary information. The GGNNs for the different interests are used to generate the interest-based node embedding of each item, which can model the complex item transition patterns for a specific interest as follows:

$$\mathbf{v}_m^{c,K} = GGNN_m\left(\mathbf{v}_c, \mathbf{A}'_{c,m}\right) \tag{12}$$

where $\mathbf{v}_m^{c,K} = \{\mathbf{v}_{1,m}^{c,K}, \ldots, \mathbf{v}_{n,m}^{c,K}\}$ contains the interest-based node embeddings of items in the current sequence, which are used to generate the interest-based current representations.

3.4 Interest-based Representations

Most session-based recommendation methods capture the main interest in each session and use a single latent vector to represent the session context. By contrast, we attempted to model the various interests in each session and generate multiple vectors for different interests. Information on multiple interests might lead to recommendation diversity and be more useful than information on the main interest only.

3.4.1 Interest-based Current Representations. The interest-based node embedding $\mathbf{v}_m^c, t$ is used to generate an interest-based current representation, which consists of three types of session embeddings: an interest-based session embedding, a global preference session embedding, and a local preference session embedding.

Interest-based Session Embedding. For the items in the current sequence, the relation scores between different items and interests are different. To account for the relation scores between items and interests, the interest attentive method is used to determine the weighted sum of the node embeddings for a specific interest to obtain the interest-based session embedding. A vector of weights is generated, and the weighted sum is determined to obtain the interest-based session embedding of the m^{th} interest as follows:

$$\mathbf{a}^m = \operatorname{softmax}\left(\mathbf{w}_2^{m\top}\left(\mathbf{W}_1^m \mathbf{v}_m^{c,K}\right)\right)^\top \tag{13}$$

$$\mathbf{s}_{\mathrm{I}}^m = \sum_{i=1}^n \mathbf{a}_i^m \mathbf{v}_{i,m}^{c,K} \tag{14}$$

where $\mathbf{v}_m^{c,K}$ is the set of interest-based node embeddings in the current sequence for the m^{th} interest, $\mathbf{a}^m \in \mathbb{R}^n$ is a vector of weights for the m^{th} interest, $\mathbf{W}_1^m \in \mathbb{R}^{d\times d}$ and $\mathbf{w}_2 \in \mathbb{R}^d$ are the projection matrices for the m^{th} interest, the superscript $\top$ represents the transpose of the matrix or vector, and $\mathbf{s}_I^m \in \mathbb{R}^d$ denotes the interest-based session embedding for the m^{th} interest.

Global Preference Session Embedding. The global preference session embedding is the last-click-based session embedding. All the attention scores in a session for the last-clicked item are computed as follows to obtain the weighted sum of all the node embeddings as a global preference session embedding:

$$\mathbf{a}_i^m = \mathbf{q}^\top \sigma\left(\mathbf{W}_1 \mathbf{v}_{n,m}^{c,K} + \mathbf{W}_2 \mathbf{v}_{i,m}^{c,K} + \mathbf{b}\right) \tag{15}$$

$$\mathbf{s}_{\mathrm{g}}^m = \sum_{i=1}^{n} \mathbf{a}_i^m \mathbf{v}_{i,m}^{c,K} \tag{16}$$

where $\mathbf{s}_g^m \in \mathbb{R}^d$ denotes the global session embedding for the m^{th} interest. The parameters $\mathbf{q} \in \mathbb{R}^d$ and $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d\times d}$ are the weights for adjusting the attention scores.

Local Preference Session Embedding. Because a user's action may be closely related to their previous actions, the last item in the current sequence that the user interacted with indicates their instant preference. Therefore, the last-click item node embedding is used to represent the local preference of a sequence as follows:

$$\mathbf{s}_l^m = \mathbf{v}_{n,m}^{c,K} \tag{17}$$

where $\mathbf{s}_l^m \in \mathbb{R}^d$ represents the local preference session embedding for the m^{th} interest.

Finally, the interest-based session embedding, global preference session embedding, and local preference session embedding are summed to acquire the summation vector, which is transformed using $\mathbf{W}_3 \in \mathbb{R}^{d\times 2d}$ to obtain the final interest-based current representation of the m^{th} interest.

$$\mathbf{c}^m = \mathbf{W}_3\left(\mathbf{s}_I^m + \mathbf{s}_g^m + \mathbf{s}_l^m\right) \tag{18}$$

where $\mathbf{c}^m \in \mathbb{R}^d$ is the interest-based current representation of the m^{th} interest and $\mathbf{W}_3 \in \mathbb{R}^d$ is the projection matrix of the summation vector.

3.4.2 Interest-based Historical Representations. By using the interest-based historical sequence embedding $\mathbf{e}_m = \{\mathbf{e}_{1,m}, \ldots, \mathbf{e}_{n,m}\}$, the self-attention method is implemented with the interest-based current representation $\mathbf{c}^m$ as the query, which ensures that the historical interest is consistent with the current interest. Because the self-attention mechanism uses historical interests to predict the current interest, the predicted current interest is consistent with the historical interests.

$$\mathbf{h}^m = \mathrm{softmax}\left(\frac{\left(\mathbf{c}^m\mathbf{W}^Q\right)\left(\mathbf{e}_m\mathbf{W}^K\right)^T}{\sqrt{d}}\right)\left(\mathbf{e}_m\mathbf{W}^V\right) \tag{19}$$

where $\mathbf{h}^m \in \mathbb{R}^d$ is the interest-based historical representation for the m^{th} interest and $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d\times d}$ are the projection matrices of the query, key, and value, respectively.

3.5 Prediction And Optimization

To account for the overall interest in both the current and historical contexts, the interest-based current and historical representations are combined as follows:

$$\mathbf{C}^m = \mathbf{W}_4\left(\mathbf{c}^m + \mathbf{h}^m\right) \tag{20}$$

where $\mathbf{C}^m \in \mathbb{R}^d$ is the context representation for the m^{th} interest and $\mathbf{W}_4 \in \mathbb{R}^{d\times d}$ is the projection matrix of the summation of the interest-based current representation $\mathbf{c}^m$ and historical representation $\mathbf{h}^m$.

After the interest-based context representation $\mathbf{C} = \{\mathbf{C}^1, \dots, \mathbf{C}^k\}$ are obtained, they are integrated to generate the target-aware context representation $\mathbf{S}_t$. IINs are used to compute the relation weights between target items and interests; however, in the IINs, the Gumbel-softmax function is replaced by the softmax function. The relation weights are used to obtain the weighted sum of $\mathbf{C}$ to generate $\mathbf{S}_t$, which has a high probability that the target item v_t is output.

$$\mathbf{a}_{t,m} = \mathbf{v}_t^\top \mathbf{W}_I\left[:, m\right] \tag{21}$$

$$\omega_{t,m} = \frac{\exp\left(\mathbf{a}_{t,m}\right)}{\sum_{h=1}^{k} \exp\left(\mathbf{a}_{t,h}\right)} \tag{22}$$

$$\mathbf{S}_t = \sum_{h=1}^{k} \omega_{t,h} * \mathbf{C}^h \tag{23}$$

where $\omega_{t,m}$ is the correlation weight of target item v_t for the m^{th} interest.

The target-aware context representation $\mathbf{S}_t$ and target item embedding $\mathbf{v}_t$ are then fed into the output layer to generate the recommendation score of the target item v_t. The higher the relevance of a target item to the context representation, the higher is the recommendation score. The inner product is determined to predict the degree of relevance between a target item and the context representation for obtaining the recommendation score.

$$\hat{\mathbf{z}}_t = \mathbf{S}_t \mathbf{v}_t \tag{24}$$

Finally, the probabilities ($\hat{\mathbf{y}}$) of the target items being the next item are computed as follows using the softmax function:

$$\hat{\mathbf{y}} = softmax\left(\hat{\mathbf{z}}\right) \tag{25}$$

According to the probabilities ($\hat{\mathbf{y}}$) obtained for all the target items, the top K items with the highest probabilities are added to the recommendation list for the user.

After obtaining the probabilities ($\hat{\mathbf{y}}$) of every target item, these probabilities can be used with the true target labels ($\mathbf{y}$) to compute the loss function $\mathcal{L}$ and update the model parameters during the training process. In the recommendation task, which is a classification problem, the cross-entropy loss function is used on the predicted probabilities ($\hat{\mathbf{y}}$) and true target labels ($\mathbf{y}$) to obtain high probabilities for the true target item by adjusting the model parameters. The cross-entropy loss function is expressed as follows:

$$\mathcal{L}(\hat{\mathbf{y}}) = -\sum_{i=1}^{|V|} \mathbf{y}_i \log\left(\hat{\mathbf{y}}_i\right) + \left(1 - \mathbf{y}_i\right) \log\left(1 - \hat{\mathbf{y}}_i\right) \tag{26}$$

After determining the loss of each batch of training samples, the backpropagation through time algorithm is used to train the developed model.

4 EXPERIMENTS

This section first introduces the real-world data sets used to conduct experiments and briefly describes each baseline method. Next, it presents the evaluation metrics for measuring the predictive accuracy. This section then presents a comparison of the performance of the proposed method with other methods.

We conducted extensive experiments to answer the following three research questions:

- **RQ1:** How does the performance of the proposed method compare with that of the considered baseline methods?
- **RQ2:** How does the number of interests affect the recommendation performance?
- **RQ3:** How does the multi-interest structure of the current and historical sequences affect the results of the proposed method?

The experimental results indicate that the proposed method with the multi-interest structure outperformed the baseline methods with the single-interest structure on the three collected real-world data sets. We verified that the proposed model could effectively capture the multi-interest context from a sequence. We also determined the suitable settings for the number of interests to be considered for the three data sets. The experimental results verified the importance of modeling the multi-interest context from both current and historical sequences to enhance the ability of a model to extract comprehensive user interests.

4.1 Data sets

We implemented the proposed model on three real-world data sets, namely those from Jdata, Trivago, and Reddit, which are representative data sets used for session-based recommendation.

- **Jdata**: The Jdata data set was collected from JD.com, which is a famous e-commerce website. This data set covers different types of user behaviors related to their interaction with items, including browsing, purchasing, and commenting.
- **Trivago**: The Trivago data set contains the anonymous interactions of users visiting the Trivago website, which is an online hotel search engine. This data set was collected during the RecSys Challenge 2019 and focuses on online travel RSs.
- **Reddit**: The Reddit data set was collected from the Reddit online forum. This data set comprises the list of subreddits that users have commented on and the timestamps of user comments. Subreddits are discussion boards generated by users that cover a specific topic.

Table 1. Statistics of the three adopted data sets

Dataset	Jdata	Trivago	Reddit
all the clicks	8179525	2578853	714598
train sessions	443108	43924	66283
test sessions	139052	4032	27480
all the items	44749	51789	9102
avg.length	14.05	53.15	7.63

First, we generated sequence-like data from the original transaction data and filtered out sequences with less than five items from the Jdata and Reddit data sets and sequences with less than three items from the Trivago data set to obtain processed data sets of appropriate size. The Trivago and Reddit data sets included the session ID and user ID, which were used to generate users' historical and current behavioral sequences. The Jdata data set did not contain sufficient information for identifying each session; therefore, we manually divided the sequences in this data set into portions corresponding to 1 day. Because the main purpose of the proposed method is to capture multiple interests from a sequence, we used relatively long sequences. Therefore, for the Jdata, Reddit, and Trivago data sets, at least four, four, and two items, respectively, were used as the input of the current sequence. Moreover, the last item was the target item. Items occurring less than 100, 50, and 5 times in the Jdata, Trivago, and Reddit data sets, respectively, were eliminated from the sequences. For each current sequence, the number of corresponding historical sequences generated for the Jdata, Reddit, and Trivago data sets was at least three, three, and one, respectively. Subsequently, all the sequences were sorted according to their timestamps, and we divided approximately 80% and 20% of the data set into a training set and testing set, respectively. The summary statistics of the data sets are presented in Table 1.

4.2 Baselines

To evaluate the performance of the proposed method, we compared it with several well-known baseline methods, which comprised traditional approaches, deep-learning-based approaches, and GNN-based approaches. The adopted traditional approaches were POP, BPR-MF, the item-KNN algorithm, and FPMC. These approaches involve using simple techniques, such as statistical methods, similarity computing, or MF, to make predictions. The adopted deep-learning-based approaches were GRU4REC, NARM, STAMP, and SASRec. These methods involve using an RNN and attention mechanism to capture users' preferences in a sequence. The adopted GNN-based approaches were the SR-GNN, GC-SAN, FGNN, and TA-GNN, which use a GNN to model the complex item transition patterns in a session.

- **POP:** In POP, a common method for recommending popular items, the top K items that appear the most frequently in the training set are selected to form the recommendation list.
- **BPR-MF [19]:** BPR-MF is an MF-based method in which pairwise ranking loss is used as the loss function to train and optimize the model parameters.
- **Item-KNN [23]:** The item-KNN algorithm recommends items that are similar to the items in the current session by computing the cosine similarity between items.
- **FPMC [20]:** In FPMC, the concept of an MC is combined with MF to conduct sequential recommendation.
- **GRU4REC [7]:** GRU4REC is an RNN-based method that captures the general interest in a s ession.
- **NARM [11]:** NARM is a model with an encoder–decoder structure, where the encoder is responsible for capturing the main interest in a session.
- **STAMP [14]:** STAMP captures the general interest and current interest within a session for making recommendations.
- **SASRec [8]:** SASRec is a self-attention-based model that can capture long-term semantics and focus on important actions.
- **SR-GNN [40]:** The SR-GNN constructs a session graph to model complex item transition patterns.
- **GC-SAN [41]:** In the GC-SAN, a GGNN is combined with the self-attention mechanism to extract much information from session data.
- **FGNN [16]**: The FGNN uses graph attention networks to model the different importance levels of neighboring nodes and constructs the session embedding.
- **TA-GNN [42]** : In the TA-GNN, the target item is modeled using target-aware attention with a GGNN.
- **LESSR**: The LESSR contains an edge-order-preserving aggregation layer and a shortcut graph attention layer for handling information loss problems.

4.3 Evaluation Metrics and Parameter Setup

We used the following evaluation metrics to evaluate the recommendation performance of the proposed method against the aforementioned methods. We selected K items from a large item pool to generate a top-K recommendation list. The parameter K was set as 20 for computing the evaluation metrics.

- **P@K** (precision)

  Precision is a commonly used measure of the predictive accuracy. It indicates the proportion of true target items occurring in the top-K recommendation list and is defined as follows.

$$\text{P@K} = \frac{n_{\text{hit}}}{N} \tag{27}$$

  where N is the total number of sessions and n_{hit} is the number of true target items in the top-K recommendation list.

- **MRR@K** (mean reciprocal rank)

The mean reciprocal rank (MRR) indicates not only the correctness of recommendation but also the recommendation ranking in the top-K recommendation list.

$$\mathrm{MRR@K} = \frac{1}{N} \sum_{v_{\mathrm{label}} \in S_{\mathrm{test}}} \frac{1}{\mathrm{Rank}\ (v_{\mathrm{label}})} \tag{28}$$

where Rank(v) is the ranking of item v in the top-K recommendation list.

- **NDCG@K** (normalized discounted cumulative gain)

  The normalized discounted cumulative gain (NDCG) accounts for the positions of the true items in the recommendation list. A score is ignored if the corresponding true item is ranked behind with the $\log_2$ function, and vice versa.

$$\mathrm{NDCG@}K = \frac{1}{Z} \sum_{i=1}^{K} \frac{2^{r_i} - 1}{\log_2(i+1)} \tag{29}$$

  where $r_i \in \{0, 1\}$ is the true label of the item in the ith position of the recommendation list and Z is the normalization term.

We set the number of dimensions of the embedding vectors (d) to 100. We set the number of interests to 7 for Jdata and 9 for both Trivago and Reddit data sets by tuning the testing data, which led to the best results in every training–testing process. All the parameters were initialized using a Gaussian distribution with a mean of 0 and a standard deviation of 0.1 to ensure that all the computing was conducted in the same range. We selected the Adam optimizer to conduct training with a batch size of 30 for Jdata and Reddit data sets, 10 for Trivago dataset. The initial learning rate was 0.001, and it decreased by 0.1 every three epochs. In addition, we set the $L2$ penalty of weights as 0.00001 to avoid inappropriate weight values.

Table 2. Performance of proposed method and baseline methods for three real-world data sets.

Method	Jdata			Trivago			Reddit		
	R@20	MRR@20	NDCG@20	R@20	MRR@20	NDCG@20	R@20	MRR@20	NDCG@20
POP	2.98	0.75	1.43	0.74	0.29	0.42	31.67	12.08	18.15
BPRMF	25.41	13.36	18.38	63.09	51.9	59.02	57.42	30.21	43.25
Item-KNN	26.77	9.62	15.43	5.58	1.54	2.77	33.96	12.49	19.7
FPMC	24.24	11.85	16.34	66.64	58.85	62.83	47.66	22.89	32.1
GRU4REC	31.08	11.39	18.19	34.52	12.82	20.3	44.79	22.29	31.24
NARM	31.54	12.23	16.5	78.62	69.41	73.63	58.56	32.32	42.23
STAMP	35.41	12.39	20.14	14.26	10.72	12.28	61.56	36.59	46.74
SASRec	19.91	6.22	9.19	68.65	42.87	48.75	61.92	36.95	42.61
SR-GNN	35.16	15.69	20.03	60.24	48.6	51.33	62.7	38.51	44.08
GC-SAN	33.56	14.11	18.41	47.47	42.71	43.8	57.25	35.71	40.59
FGNN	29.04	10.32	14.47	40.89	29.27	31.94	63.33	35.1	41.67
TA-GNN	38.62	16.8	21.70	61.08	52.32	54.39	64.55	41.18	46.55
LESSR	39.15	14.42	20.34	62.79	53.61	55.7	61.37	35.68	41.53
MI-GNN	**40.22**	**18.49**	**23.36**	**82.16**	**76.11**	**77.26**	**64.82**	**41.6**	**47**

4.4 Performance Comparison

To compare the performance of the proposed method and baseline methods, we computed the Recall@20, MRR@20, and NDCG@20. We tested the proposed model on three real-world data sets, namely those of Jdata, Trivago, and Reddit, which are widely used in recommendation experiments. The Jdata data set contains data on users' browsing behaviors on an e-commerce website, the Trivago data set comprises data on users' behaviors when searching for holiday accommodations, and the Reddit data set comprises data on users' commenting behavior on Reddit. These three data sets, which are related to different categories of online services, contain information on different types of user behavior patterns. These data sets were used to conduct experiments on the proposed model's ability to model distinct user behavior patterns. Table 2 presents all the results obtained for the proposed method and baselines methods.

As indicated in Table 2, the proposed method outperformed the baseline methods. In POP, only rudimentary statistical methods are used to determine the frequency of items in the entire data set for recommending frequently occurring items; however, this approach is too simple to model the complexities of recommendation. Unlike in POP, in which the state of the user is neglected, in BPR-MF, personalized recommendation is performed by optimizing the pairwise ranking loss; thus, the BPR-MF method outperformed POP. The aforementioned result indicates the importance of personalization in recommendation. The item-KNN algorithm simply considers the similarity between items on the basis of their co-occurrence in different sequences and does not consider sequential information. FPMC is a method based on a first-order MC, and this method only considers short-term actions and ignores the high-order dependence between items. The strict assumption of rigid order limits the prediction ability of MC-chain-based methods. The item-KNN algorithm performed significantly worse than did the other baseline methods on the Trivago data set because this data set was sparse, which made it difficult for the item-KNN algorithm to model comprehensive item similarities. By contrast, FPMC achieved superior performance on the Trivago data set, which indicates that short-term actions are crucial in next-item prediction. However, the item-KNN algorithm and FPMC exhibited unique disadvantages in modeling sequential data.

Among deep neural networks, RNNs, such as GRU4REC, NARM, and STAMP, are widely used to identify features in a sequence because they can model the sequential dependence in time-series data. GRU4REC utilizes GRUs, which are variants of RNNs, to model the general interest in a session. NARM uses two RNN-based encoders to consider both the local and global interest in a session. This strategy enables NARM to outperform GRU4REC and exhibit superior feature extraction from a session. Moreover, STAMP uses short-term memory to model the last-clicked item as the current interest. Most of the experimental results obtained for NARM and STAMP indicate the importance of modeling short-term actions in next-item prediction. Rather than using an RNN, SASRec uses the self-attention method to capture long-term semantics and recognize important previous actions in each time step. All the aforementioned methods integrate information from the entire session and model the transition patterns between successive items. However, the dependencies of items within a session are too complicated to be accurately modeled only by considering the relations between items. GNNs efficiently model complex item transition patterns and represent sessions with accurate embedding vectors; thus, GNN models have superior recommendation performance. The GNN-based methods analyzed in the experiment can capture complex item transition patterns in a session by constructing a session graph for acquiring the precise node embedding that forms the session embedding.

The SR-GNN transforms sequences into graph structures for capturing the complicated item dependence. In the GC-SAN, a GGNN is combined with the self-attention mechanism to retain relevant information and neglect noise. The FGNN considers not only the sequence order but also the latent order in a session. The TA-GNN uses target-aware attention to construct a session embedding that has a high probability of the target item being target. The TA-GNN outperformed the other GNN-based methods on all the adopted data sets, which indicates the importance of considering the target items in recommendation generation. GNN-based methods performed well in session recommendation; however, they always aggregated various types of information in sequence by obtaining representations of a single session, which fail to represent the different contexts related to different interests in a sequence. The proposed method possesses the advantages of GNN-based methods, and it involves the extraction of the multi-interest context of a sequence; thus, the proposed method outperformed the baseline methods in this study.

As is the case with most GNN-based method, a session graph is constructed in the proposed method to capture complicated item transition patterns. The SR-GNN considers the last-clicked item as a query to form the session embedding of general preference, whereas the TA-GNN generates a target-aware embedding that reflects the user's preference toward the target item. Moreover, the LESSR constructs directed multigraphs for solving the lossy session encoding problem and a shortcut graph to model the long-range dependencies between sequence items. Some models, such as the H-RNN and HierTCN models, model both long- and short-term user interests. For the Reddit data set, the Recall@20 and MRR@20 values of the H-RNN model were 61.8% and 33.88%, respectively, and the Recall@20 and MRR@20 values of the HierTCN model were 63.96% and 34.27%, respectively. The aforementioned models do not consider the multi-interest context of behavior sequences. Therefore, their performance was inferior to that of the MI-GNN model, which captures multiple user interests from users' long- and short-term behaviors. However, the multiple interests in a sequence are not identified in the aforementioned two methods. Rather than aggregating various types of information without considering the variety of interests involved, interest graphs are constructed in the proposed method for effectively extracting the correlation between different items and interests; thus, the proposed method outperformed the baseline methods in this study.

4.5 Analysis on the Multi-Interest Structure

The proposed method mainly involves capturing the multi-interest context from sequence data. Therefore, we examined the results obtained under different numbers of interests. The results indicate that the multi-interest structure outperformed the single-interest structure. We attribute this result to the fact that the multi-interest structure can capture comprehensive features from users' behavior sequences and avoid missing information. Extensive experiments were performed using the proposed method by setting the number of interests k as $\{1, 3, 5, 7, 9\}$. Figure 6 displays the results obtained with the proposed method on the three adopted data sets for different numbers of interests. For the Jdata data set, the proposed method achieved the best performance on all the evaluation matrices when the number of interests was set as 7. When increasing the number of interests to 9, the performance of the proposed method marginally declined; thus, a higher number of interests did not necessarily lead to superior performance. For the Trivago data set, the best recall was achieved when the number of interests was 7; however, the best MRR and NDCG were achieved when the number of interests was 9. An increase in the number of considered interests enhanced the ability of the proposed model to distinguish between the many interests of the user, which resulted in the model generating a more precise recommendation list with items ranked according to the probabilities of user preferences. However, an increase in the number of considered interests decreased the proposed model's ability to select the correct items into the top-K recommendation list. For the Reddit data set, the model with the single-interest structure marginally outperformed all the models with the multi-interest structure on Recall@50 but was significantly outperformed by the models with the multi-interest structure on the MRR and NDCG. The model with three interests significantly outperformed the model with the single-interest structure in terms of the MRR and NDCG. A negative correlation was observed between recall and MRR (NDCG). The model with the single-interest structure can select the target item into a recommendation list more accurately than it can rank the target items. By contrast, the models with the multi-interest structure can rank items in a recommendation list in the precise order. Thus, these models yielded higher MRR and NDCG values than did the model with the single-interest structure. The Reddit data set contained the least number of target items among the adopted data sets. When the model with a single-interest structure is employed, selecting

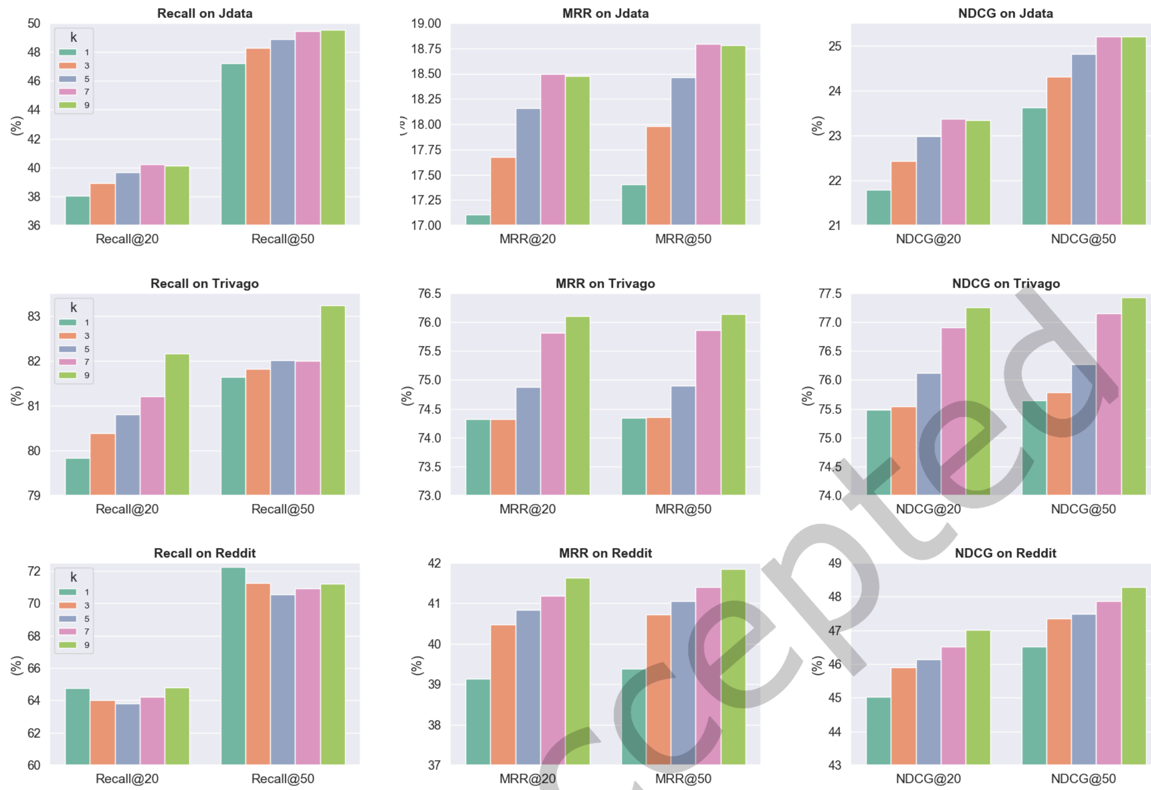

Fig. 6. Performance of proposed method on the three adopted data sets for different numbers of interests.

the correct target item into the recommendation list is easier than is ranking the target items correctly. Therefore, the model with a single-interest structure marginally outperformed the models with a multi-interest structure on Recall@50 but was considerably outperformed by these models on the MRR and NDCG. This result indicates that the multi-interest structure helped the models rank items in the recommendation list in the precise order.

With regard to the proposed method, the multi-interest structure exhibited a superior performance to the single-interest structure. The multi-interest structure could extract the multi-interest context from a sequence and represent it with multiple vectors to prevent the missing of information. However, the consideration of a higher number of interests did not necessarily lead to superior performance. The best setting for the number of interests was not the same for the three data sets. This result can be attributed to the unique characteristics of the data sets. Some data sets are complex and contain a wide range of items. The user behavior sequences from these data sets might comprise various items that represent multiple interests and desires. The modeling of such diverse interests requires a multiple-interest model. Some data sets are relatively simple, and their user behavior sequences contain rather similar items. Therefore, models with an excessive number of interests are not required to model these sequences. Moreover, an excessively complicated model may exhibit poor performance for simple sequences because of data set overexploitation. Therefore, identifying the appropriate multi-interest structure for different data sets is crucial. Considering the experimental results, we set the number of interests (k) to 7 when modeling the Jdata data set and 9 when modeling the Trivago and Reddit data sets to achieve the best performance.

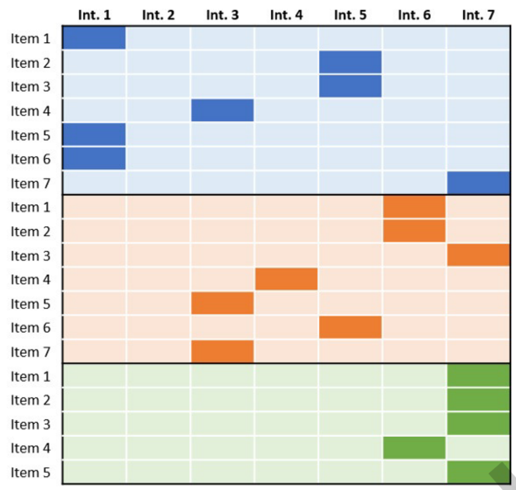

Fig. 7. Visualization of interest learned from different users for the Jdata data set.

In addition, we analyzed how interests are learned by visualizing the results of interest identification networks for users' historical behavior sequences. According to equation (2), each item in a historical behavior sequence is associated with a specific interest. Figure 7 displays the historical behavior sequences of a user for the Jdata data set. Areas with different colors in this figure represent distinct historical behavior sequences. The user's first historical behavior sequence mostly comprised items related to the first and fifth interests. In other words, this user may have two kinds of purchasing needs. With time, the user stopped focusing on the items related to the first interest, as indicated by their second and third historical behavior sequences. The second sequence contained items related to various interests, and the third sequence mainly included items related to the seventh interest. Through the visualization of interest weight, we can understand users' shopping habit more easily.

Table 3. Performance of single-interest and multi-interest methods.

Method	Jdata			Trivago			Reddit		
	R@20	MRR@20	NDCG@20	R@20	MRR@20	NDCG@20	R@20	MRR@20	NDCG@20
SASRec	19.91	6.22	9.19	68.65	42.87	48.75	61.92	36.95	42.61
SR-GNN	35.16	15.69	20.03	60.24	48.60	51.33	62.7	38.51	44.08
TA-GNN	38.62	16.8	21.70	61.08	52.32	54.39	64.55	41.18	46.55
MIND	36.96	13.26	18.57	78.57	51.43	57.92	**68.47**	35.13	43.03
ComiRec-SA	37.05	13.53	18.8	77.87	51.99	58.11	68.14	31.81	40.33
ComiRec-DR	36.38	13.91	18.92	76.26	54.3	59.5	64.03	34.52	41.42
MI-GNN	**40.22**	**18.49**	**23.36**	**82.16**	**76.11**	**77.26**	64.82	**41.64**	**47**

We compared our best results with those of some prominent methods with single- and multi-interest structures. For a fair comparison, the numbers of interests in the compared multi-interest methods were the same as those in the best settings of the proposed method for the three data sets. The results obtained for the compared methods are presented in Table 3. We selected the following three methods with the multi-interest structure as baselines:

- **MIND:** MIND utilizes the dynamic routing mechanism to extract users' various interests and encode them with multiple representations.
- **ComiRec-SA:** ComiRec-SA uses the self-attention mechanism, which involves performing the attention operation multiple times, to capture the multi-interest context.
- **ComiRec-DR:** ComiRec-DR uses the dynamic routing method (similar to CapsNet [21]) to capture the sequential information and multiple interests from a sequence.

The proposed method outperformed the other methods with the single- or multi-interest structure in terms of all the evaluation metrics for all the data sets except Recall@20 for Reddit. For the Jdata data set, when the target-aware embedding was used to consider the target item, the TA-GNN outperformed the baseline methods. However, all the models with the multi-interest structure outperformed SASRec and SR-GNN; thus, extracting multiple interests from a sequence is still crucial. The performance of ComiRec-SA was superior to that of SASRec, which indicates that the multi-interest-based self-attentive method outperforms the multihead self-attention mechanism. The results of two dynamic-routing-based methods, namely MIND and ComiRec-DR, did not significantly differ. The Trivago data set comprised data on users' hotel searching behavior and thus is different from the e-commerce data set containing data on distinct behavioral patterns. Thus, different results were obtained for the Trivago and Jdata data sets. All the multi-interest-based methods, including the proposed method, outperformed the single-interest-based methods on most of the evaluation metrics on the Trivago data set. The Trivago data set contained the highest number of target items and exhibited the highest average session length among the adopted data sets. Thus, the Trivago data set represented complex user behaviors. Consequently, the multi-interest context was crucial for the sequences of this data set. For the Reddit data set, MIND achieved the best performance in terms of Recall@20 but exhibited relatively poor results in terms of MRR@20 and NDCG@20. The proposed method, which involved precisely modeling the multi-interest context by using a GNN-based method, generated a recommendation list with a more accurate item ranking than did MIND.

4.6 Analysis of Modeling of Multi-Interest Context in Current and Historical Sequences

We compared the efficiencies of modeling the multi-interest context in current and historical sequences and found that the models that considered historical context exhibited superior results. Extensive experiments were performed with the following models:

- **MI-GNN_MI-C**: This model only considers the multi-interest context in the current sequence.
- **MI-GNN_H_C**: This model considers both historical sequences and the current sequence with a single-interest structure.
- **MI-GNN_H_MI-C**: This model captures the multi-interest context in the current sequence but models the historical sequences with a single-interest structure.
- **MI-GNN_MI-H_C**: This model captures the multi-interest context from historical sequences but models the current sequence with a single-interest structure.
- **MI-GNN(-TA)**: This model captures the multi-interest context from the current sequence and historical sequences without using a target-aware attention mechanism.
- **MI-GNN**: This model is the proposed model, which captures the multi-interest context in the current sequence and historical sequences.

Table 4 presents a comparison of the results obtained with the proposed model, namely the MI-GNN, and the other four aforementioned models. MI-GNN_MI-H_C performed the worst among the five models compared, which indicates that capturing the multi-interest context from the current sequence is essential to providing accurate recommendations. The current sequence reflects the multiple instant interests of a user, and the next item that a user may interact with is highly related to these interests. The proposed model extracted multiple

Table 4. Performance of models with single- and multi-interest frameworks that model the current sequence and historical sequences.

Method	Jdata					
	R@20	MRR@20	NDCG@20	R@50	MRR@50	NDCG@50
MI-GNN_MI-C	39.55	17.98	22.81	48.84	18.28	24.66
MI-GNN_H_C	38.02	17.10	21.78	47.21	17.4	23.61
MI-GNN_H_MI-C	40.15	18.32	23.22	**49.56**	18.62	25.09
MI-GNN_MI-H_C	37.46	16.81	21.44	46.61	17.11	23.26
MI-GNN(-TA)	37.76	17.15	21.77	47.07	17.45	23.62
MI-GNN	**40.22**	**18.49**	**23.36**	49.45	**18.79**	**25.2**

interests from the current sequence to generate a precise recommendation list that satisfied these interests; thus, the proposed model performed better than the other models. Moreover, MI-GNN_H_C, which has a single-interest structure, outperformed MI-GNN_MI-H_C. Interest-based historical representations were generated with the self-attention method by using the current representation as a query. In MI-GNN_MI-H_C, if the current representation contains only a single interest rather than multiple interests, the multiple-interest-based historical representations are inconsistent with the interest in the current sequence, which leads to inaccurate results. Thus, in this study, historical sequences with the multi-interest structure could not provide precise information and even generated noise for the prediction. Consequently, MI-GNN_MI-H_C performed worse than did the model with only a single-interest structure.

MI-GNN_MI-C only considers the multi-interest context in the current sequence and does not consider any information from historical sequences; therefore, this model exhibited worse performance than did MI-GNN_H_MI-C. The aforementioned result indicates the importance of modeling both long- and short-term interests from across sequences in recommendation generation. MI-GNN_H_MI-C only considers the main interests in historical sequences and exhibited marginally worse performance than did the MI-GNN. Because the sequences of the Jdata data set were split manually according to a specific time interval, our model might not accurately extract the multi-interest context of historical behavior sequences; therefore, the performance of the MI-GNN model was not notably higher than that of the MI-GNN_H_MI-C model. Nevertheless, historical behavior sequences reflected the long-term evolutions in multiple user interests. For modeling comprehensive user interests from current and historical sequences, the multi-interest context should be efficiently extracted from historical sequences and the historical interests should be made consistent with the corresponding current interests. The MI-GNN(-TA) model considers current and historical behavior sequences but without using a target-aware attention mechanism; thus, it provides lower output values than does the MI-GNN model. Because different target items are related to a user's diversified interests, the user's specific interests should be activated according to distinct target items for recommendation. Thus, the MI-GNN, which extracts the multi-interest context from the current sequence and historical sequences, achieved the best performance among the five compared models. We used Python 3.7.11 and PyTorch 1.4.0 on an NVIDIA RTX 2080 graphics processing unit with 2,944 CUDA cores to implement our framework. The operating system was Ubuntu 18.04. The training time in the optimal setting was 200, 25, and 15 min per epoch for the Jdata, Reddit, and Trivago data sets, respectively.

The experimental results in Table 4 verify that the modeling of historical sequences allows the multi-interest context to be extracted from the historical sequences and enhances the ability of a model to comprehensively represent the varied interests of users. We also compared the results obtained with the proposed method for different maximum numbers of considered historical sequences. Among the three adopted data sets, the Reddit data set had the highest average number of historical sequences corresponding to a single current sequence;

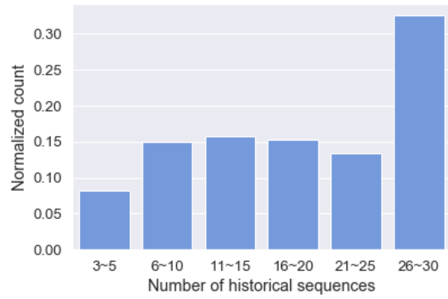

Fig. 8. Histogram of normalized counts for different numbers of historical sequences corresponding to each current sequence for the Reddit data set.

Fig. 9. Performance of the proposed method on the Reddit data set for different numbers of maximum historical sequences.

therefore, we used this data set to perform the aforementioned comparisons. Figure 8 shows the normalized counts for different numbers of historical sequences. Because we only used the 30 most recent historical sequences during data preprocessing, the range of 26–30 maximum historical sequences had the highest normalized count. Figure 9 displays the recall, MRR, and NDCG results of the proposed model with different settings for the maximum number of historical sequences. The highest recall was obtained when setting the maximum number of historical sequences as 15; however, the best MRR and NDCG were obtained when the number of historical sequences was set as 20. The results indicated that considering information from more historical sequences did not necessarily lead to better performance. Because user interests change over time, some interests reflected by the items in historical sequences might have low importance in the current sequence. Extracting information from historical sequences that represent interests with low importance in the current sequence is not necessary for predicting the next item of the current sequence. The consideration of such historical sequences may even lead to a decrease in the recommendation accuracy.

5 CONCLUSIONS

Next-item recommendation is a challenging task to predict the next item that the user might be interested in by capturing the user's interest from the sequential behaviors. Most previous methods focus on extracting the single interest from the user's browsing sequence which actually reflects the user's various interests. Consequently, the MI-GNN extracts users' multiple interests from the current sequence and historical sequences by constructing multiple-interest graphs to model the complex item dependencies for different interests. Sequential behaviors represent the varied interests of users that prompt them to browse various items; therefore, extracting the multi-interest context from sequences is crucial in recommendation generation. The current sequence reflects users' current interests, whereas historical sequences reflect the long-term evolution of user interests. The current and long-term interests of users are closely related. Therefore, the proposed method involves considering the current sequence and historical sequences to construct a current interest graph and historical interest graphs, respectively. First, a GNN-based method is used to construct interest graphs for capturing the complex item transition patterns for various interests. Subsequently, on the basis of a user's multiple interests, multiple-interest-based representations are generated to precisely reflect each interest and preventing the information loss that occurs when only the main interest of a sequence is modeled. Finally, by clearly recognizing the multiple interests from a sequence and representing them, the proposed model selects the items with the highest probabilities of being the next item to generate a recommendation list that satisfies the user's diverse interests.

The proposed method involves extracting users' multiple interests from their behavior sequence data. Moreover, a GNN, which is widely used to model the relationship between entities, was developed to consider item dependencies in terms of users' different interests. By integrating users' historical and current interests, we comprehensively modeled their evolving and diversified interests for making accurate recommendations. The results of the extensive experiments conducted in this study indicated that the proposed model with the multi-interest structure outperformed models with the single-interest structure; thus, capturing the multi-interest context from a sequence by developing interest graphs to model the complicated item transition patterns for various interests is a crucial task. Moreover, we verified that the multi-interest structure of the proposed model for both historical and current sequences is suitable. This structure helps the model correlate historical and current interests and enhances the model's ability to learn the user's comprehensive set of interests for predicting the next item. Finally, the proposed method outperforms baseline methods on three real-world data sets, which achieves 4% improvement of Recall over the SOTAs on Jdata dataset .

There are different kinds of recommender systems developed nowadays, like conversational RSs, group RSs and cross-domain RSs, which are probable to combine with our multi-interest-based method to extract more comprehensive interests from the user or the group. In cross-domain recommendation [4, 12, 44], the problem of data sparsity is overcome by extracting information from various sources related to different domains and by analyzing user preferences in other domains apart from the target domain to obtain an better understanding of user preferences. To model the multiple interests of users, learning their variety of interests in different domains and transferring these interests to the target domain is a good strategy. This strategy helps the model better capture the multi-interest context in a sequence and thus constitutes a possible improvement to our proposed method.

REFERENCES

[1] Ludovico Boratto, Salvatore Carta, and Roberto Saia. 2016. Improving the Accuracy of Latent-space-based Recommender Systems by Introducing a Cut-off Criterion.. In *EnCHIReS@ EICS*. 44–53.

[2] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable Multi-Interest Framework for Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.

[3] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 108–116.

[4] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Spanish conference on information retrieval*. sn, 1–12.

[5] Cheng Guo, Mengfei Zhang, Jinyun Fang, Jiaqi Jin, and Mao Pan. 2020. Session-based Recommendation with Hierarchical Leaping Networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1705–1708.

[6] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 549–558.

[7] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[8] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[9] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[10] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.

[11] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[12] Pan Li and Alexander Tuzhilin. 2020. DDTCDR: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 331–339.

[13] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* (2015).

[14] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1831–1839.

[15] Mehrbakhsh Nilashi, Othman Ibrahim, and Karamollah Bagherifard. 2018. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications* 92 (2018), 507–520.

[16] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 579–588.

[17] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 130–137.

[18] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4806–4813.

[19] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[20] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 811–820.

[21] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829* (2017).

[22] Roberto Saia, Ludovico Boratto, and Salvatore Carta. 2016. A semantic approach to remove incoherent items from a user profile and improve the accuracy of a recommender system. *Journal of Intelligent Information Systems* 47, 1 (2016), 111–134.

[23] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[24] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2008), 61–80.

[25] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*. Springer, 291–324.

[26] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, Sep (2005), 1265–1295.

[27] Thiago Silveira, Min Zhang, Xiao Lin, Yiqun Liu, and Shaoping Ma. 2019. How good your recommender system is? A survey on evaluations in recommendation. *International Journal of Machine Learning and Cybernetics* 10, 5 (2019), 813–831.

[28] Wenzhuo Song, Shoujin Wang, Yan Wang, and Shengsheng Wang. 2021. Next-item recommendations in short sessions. In *Fifteenth ACM Conference on Recommender Systems*. 282–291.

[29] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 17–22.

[30] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 565–573.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[32] Jinshan Wang, Qianfang Xu, Jiahuan Lei, Chaoqun Lin, and Bo Xiao. 2020. PA-GGAN: Session-Based Recommendation with Position-Aware Gated Graph Attention Network. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.

[33] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A collaborative session-based recommendation approach with parallel memory modules. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 345–354.

[34] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet A Orgun, and Defu Lian. 2021. A survey on session-based recommender systems. *ACM Computing Surveys (CSUR)* 54, 7 (2021), 1–38.

[35] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-based transactional context embedding for next-item recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[36] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2020. Intention nets: psychology-inspired user choice behavior modeling for next-basket prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6259–6266.

[37] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2020. Intention2Basket: A Neural Intention-driven Approach for Dynamic Next-basket Planning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. 2333–2339.

[38] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet A Orgun, and Longbing Cao. 2019. Modeling Multi-Purpose Sessions for Next-Item Recommendations via Mixture-Channel Purpose Routing Networks.. In *IJCAI*, Vol. 19. 3771–3777.

[39] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. 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*. 169–178.

[40] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.

[41] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation.. In *IJCAI*. 3940–3946.

[42] Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2020. TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation. *arXiv preprint arXiv:2005.02844* (2020).

[43] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 582–590.

[44] Yinan Zhang, Yong Liu, Peng Han, Chunyan Miao, Lizhen Cui, Baoli Li, and Haihong Tang. 2020. Learning personalized itemset mapping for cross-domain recommendation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2561–2567.

[45] Nengjun Zhu, Jian Cao, Xinjiang Lu, and Hui Xiong. 2021. Learning a Hierarchical Intent Model for Next-Item Recommendation. *ACM Transactions on Information Systems (TOIS)* 40, 2 (2021), 1–28.