Global Heterogeneous Graph and Target Interest Denoising for Multi-behavior Sequential Recommendation

Xuewei Li
College of Intelligence and Computing, Tianjin University
Tianjin, China
lixuewei@tju.edu.cn

Hongwei Chen
College of Intelligence and Computing, Tianjin University
Tianjin, China
chw@tju.edu.cn

Jian Yu
College of Intelligence and Computing, Tianjin University
Tianjin, China
yujian@tju.edu.cn

Mankun Zhao
College of Intelligence and Computing, Tianjin University
Tianjin, China
zmk@tju.edu.cn

Tianyi Xu
College of Intelligence and Computing, Tianjin University
Tianjin, China
tianyi.xu@tju.edu.cn

Wenbin Zhang
Information and Network Center, Tianjin University
Tianjin, China
zhangwenbin@tju.edu.cn

Mei Yu*
College of Intelligence and Computing, Tianjin University
Tianjin, China
yumei@tju.edu.cn

ABSTRACT

Multi-behavior sequential recommendation (MBSR) predicts a user's next item of interest based on their interaction history across different behavior types. Although existing studies have proposed capturing the correlation between different types of behavior, two important challenges have not been explored: i) Dealing with heterogeneous item transitions (both global and local perspectives). ii) Mitigating the issue of noise that arises from the incorporation of auxiliary behaviors. To address these issues, we propose a novel solution, **G**lobal **H**eterogeneous Graph and **T**arget **I**nterest **D**enoising for Multi-behavior Sequential Recommendation (GHTID). In particular, we view the transitions between behavior types of items as different relationships and propose two heterogeneous graphs. By considering the relationship between items under different behavioral types of transformations, we propose two heterogeneous graph convolution modules and explicitly learn heterogeneous item transitions. Moreover, we utilize two attention networks to integrate long-term and short-term interests associated with the target behavior to alleviate the noisy interference of auxiliary behaviors. Extensive experiments on four real-world datasets demonstrate that our method outperforms other state-of-the-art methods.

*Corresponding author.

WSDM '24, March 4–8, 2024, Merida, Mexico.

ACM ISBN 979-8-4007-0371-3/24/03…$15.00
https://doi.org/10.1145/3616855.3635857

CCS CONCEPTS

• **Information systems** → **Information systems applications**.

KEYWORDS

Multi-Behavior Sequential Recommendation, Graph Neural Network, User Interest Denosing

ACM Reference Format:
Xuewei Li, Hongwei Chen, Jian Yu, Mankun Zhao, Tianyi Xu, Wenbin Zhang, and Mei Yu. 2024. Global Heterogeneous Graph and Target Interest Denoising for Multi-behavior Sequential Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), March 4–8, 2024, Merida, Mexico.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3616855.3635857

1 INTRODUCTION

As information technology advances, the problem of information overload becomes increasingly severe. To mitigate this issue, recommendation systems have been widely developed. Among them, sequential recommendation systems (SRS) have received considerable attention, as it predicts the next item of interest by learning the sequential features of a user's historical items. Current research on SRS mainly focuses on contrastive learning[27], item attributes[28], and multiple interests[22], among others, to further improve the recommendation effect.

Despite the validity of the current approaches, most research has only focused on a single type of interaction and overlooked the naturally occurring multi-behavioral interactions. In practical recommendation scenarios, different behaviors exist between users and items. For example, on shopping websites, there are various behaviors between users and items, such as Page View (PV) and Purchase (Pur), as illustrated in Figure1. Pur, as the target behavior, reflects the user's actual interest, while PV, as an auxiliary behavior,

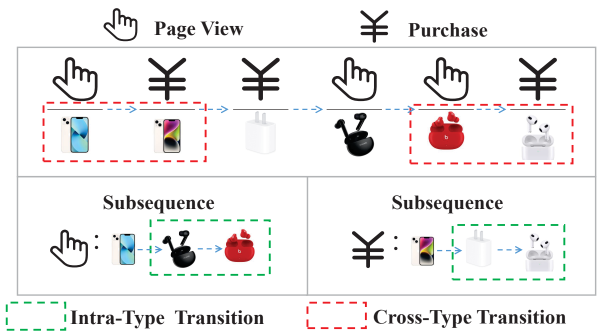

Figure 1: Illustration of heterogeneous item transitions, including intra-type transition and cross-type transition.

can provide additional information to help predict the target behavior. Therefore, when incorporating multiple types of behavior into a sequential model, there are two main challenges to consider.

- **Dealing with heterogeneous item transitions.** There may be complex transitions between items of different behaviors. For example, a user views (Page View, PV) and purchases (Pur) some items, we can categorize the item sequence into sub-sequences by behavior type. The items in different sub-sequences have different relationships, such as from PV to PV, from Pur to Pur. In addition, there are also cross-behavior type relationships between two items in the user's item sequence, such as from PV to Pur, from Pur to PV. We define this relationship as heterogeneous item transitions, including item transitions between the same type of behavior (intro-type transition) and item transitions between different types (cross-type transition), as shown in Figure 1. These transitions reveal a user's individual behavior patterns and preferences, as well as the broader association rules of items across behaviors. However, most existing methods either do not explicitly model heterogeneous item transitions or ignore information from global users, and thus cannot learn heterogeneous item transitions.
- **Interference of auxiliary behaviors.** Before performing the target behavior (Pur), users may engage in a large number of auxiliary behaviors (PV) that may contain information that is irrelevant or even contrary to their final interests. For example, users may browse many items but only be interested in a few. Therefore, extracting specific interests for the target behavior from the auxiliary behaviors and filtering out the noise information in the auxiliary behaviors is another critical challenge.

To overcome the above challenges, we propose a new solution named **G**lobal **H**eterogeneous Graph and **T**arget **I**nterest **D**enoising for Multi-behavior Sequential Recommendation, termed as **GHTID**. We introduced a global item-item co-occurrence graph to explore item correlations under different behavior types and designed a global graph convolutional layer with attention mechanisms for behavior pair-specific aggregation and relationship perception. We also constructed a local item-item transition graph for each user to distinguish user-specific behavior patterns and interest preferences, and designed local convolutional layers with attention mechanisms for behavior transitions perception. These graph neural networks learn heterogeneous item transitions from global and local perspectives to generate global and personalized project representations. To reduce the interference of auxiliary behavior on the target behavior, we designed an interest aggregation module that utilizes a soft attention mechanism to learn the representations of short-term and long-term interests under the target behavior.

Overall, our main contributions are summarized as follows:

- We have customized a global item-item co-occurrence graph, local item-item transition graphs, and corresponding graph convolutional networks based on the proposed heterogeneous item transitions.
- We have proposed interest aggregation module, which effectively reduces the interference of auxiliary behaviors on the prediction of target behaviors, and further experiment demonstrates the effectiveness of this module in section 4.4.
- We conducted extensive experiments on four real datasets, demonstrating that our proposed GHTID is better than the latest multi-behavior sequential recommendation method.

2 PRELIMINARIES

In this section, we first presented the problem formulation proposed for this study. Then, we introduced two graph models proposed to learn the behavior dependency at the item level: global item-item co-occurrence graph and local item-item transition graphs, respectively.

2.1 Problem Formulation

In MBSR scenario, we have a set of $|\mathcal{U}|$ users $\mathcal{U} = \{u_1, ..., u_i, ..., u_{|\mathcal{U}|}\}$ and a set of $|\mathcal{V}|$ items $\mathcal{V} = \{v_1, ..., v_j, ..., v_{|\mathcal{V}|}\}$. We further define $\mathcal{B}$ to represent $|\mathcal{B}|$ types of different interaction behaviors $\mathcal{B} = \{b_1, ..., b_k, ..., b_{|\mathcal{B}|}\}$, such as page view, add to favorite, add to cart and purchase. For each user $u_i \in \mathcal{U}$, we use a list $S_i = [(v_{i,1}, b_{i,1}), (v_{i,2}, b_{i,2}), ..., (v_{i,n}, b_{i,n})]$ to denote his interacted sequence, where $v_{i,j} \in \mathcal{V}$, $b_{i,j} \in \mathcal{B}$ are the j-th interacted item, behavior of user u_i, and n is the preset maximum length of the sequence. We divide the behavior set $\mathcal{B}$ into a target behavior and multiple auxiliary behaviors. Specifically, We define the most valuable behavior as the target behavior (e.g., purchasing in online retail e-commerce platforms, downloading in application stores). The task of our study can be expressed as:

Input: The multi-behavior interaction sequence of each user $u_i \in \mathcal{U}$, $S_i = [(v_{i,1}, b_{i,1}), (v_{i,2}, b_{i,2}), ..., (v_{i,n}, b_{i,n})]$.

Output: The probability that u_i interacts with an item $v_j \in \mathcal{V}$ under the target behavior at time step $n + 1$.

2.2 Global Item-Item Co-occurrence Graph

To model the co-occurrence relationships between items under different behaviors from a global perspective, we define a global item-item co-occurrence graph $\mathcal{G}_g = \{\mathcal{V}_g, \mathcal{E}_g\}$.

The graph contains all item nodes, i.e., $\mathcal{V}_g = \mathcal{V}$. Let $\mathcal{R}_g = \{x \rightarrow y | x, y \in \mathcal{B}\}$ denote multiple relationships for the global graph, and each relationship $r \in \mathcal{R}_g$ reflects the correlation between items under different behaviors. Formally, the edge set is represented as $\mathcal{E}_g = \{(v_i, v_j, r, w_{ij}^r) | v_i, v_j \in \mathcal{V}, r \in \mathcal{R}_g, w_{ij}^r \in \mathbb{R}\}$, Where v_i denotes the source node, v_j denotes the target node, r represents the transition type between the two items, and w_{ij}^r represents the co-occurrence coefficient of the two item nodes v_i and v_j on the relationship r. We propose improving point-wise mutual information (PMI)[30] to measure the co-occurrence coefficient between items under different relationships. The PMI of item nodes v_i and v_j under the relationship $x \longrightarrow y$ is defined as follows:

$$
\begin{aligned}
\mathrm{PMI}(v_i, v_j, x \longrightarrow y) &= \log \frac{\mathrm{p}(v_i, v_j, x \longrightarrow y)}{\mathrm{p}(v_i, x)\mathrm{p}(v_j, y)} \\
\mathrm{p}(v_i, v_j, x \longrightarrow y) &= \frac{|\mathcal{S}(v_i, v_j, x \longrightarrow y)|}{|\mathcal{S}|} \\
\mathrm{p}(v_i, x) &= \frac{|\mathcal{S}(v_i, x)|}{|\mathcal{S}|}
\end{aligned}
\tag{1}
$$

where $\mathcal{S}$ is the set of all item sequences, $\mathcal{S}(v_i, x)$ is the set of item sequences containing item v_i with behavior type x, and $\mathcal{S}(v_i, v_j, x \longrightarrow y)$ is the set of item sequences containing both item v_i with behavior type x and item v_j with behavior type y. The co-occurrence coefficient of v_i and v_j under the relationship $x \longrightarrow y$ is $w_{ij}^{x \rightarrow y} = \mathrm{PMI}(v_i, v_j, x \longrightarrow y)$. We only retain edges with positive PMI, as negative PMI values indicate a weak correlation between the two items, i.e., $w_{ij}^r > 0$. In order to control the complexity, we divide the item sequence into multiple sliding windows of the same size 20 for each user.

2.3 Local Item-Item Transition Graph

The graph model can flexibly represent the relationship between any two items in an item sequence. Specifically, given an interaction sequence S, we transform it into a graph $\mathcal{G}_s = \{\mathcal{V}_s, \mathcal{E}_s\}$. $\mathcal{V}_s \subset \mathcal{V}$ denotes an item set appearing in the interaction sequence S. In the local item-item transition graph, we further consider the orders between item pairs and define the set of relationships as $\mathcal{R}_s = \{x \xrightarrow{z} y | x, y \in \mathcal{B}, z \in \{+, -\}\}$. For example, when the relationship between item v_i and item v_j is $x \xrightarrow{+} y$, it indicates that the behavior type of item v_i is x, the behavior type of v_j is y, and '+' represents that the user first interacts with item v_i, and then interacts with item v_j. We define the set of edges as $\mathcal{E}_s = \{(v_i, v_j, r) | v_i, v_j \in \mathcal{V}_s, r \in \mathcal{R}_s\}$, where v_i and v_j are the source and target nodes, respectively, s is the relationship between the two nodes.

3 METHODOLOGY

This section proposes the **GHTID** model, which captures the heterogeneous item transitions and reduces the interference of auxiliary behavior. The overall framework of the model is shown in Figure 2. **GHTID** consists of three main modules: (1) Global Graph Convolution Module, which includes item-to-relationship and relationship-to-item information propagation stages, for encoding item representations. (2) Local Graph Convolution Module, which includes a behavior-aware attention mechanism for encoding local item representations (3) The Interest Aggregation Module includes target short-term interest aggregation, target long-term interest aggregation, and interest fusion for extracting the final representation of user interest.

3.1 Global Graph Convolution Module

The global graph convolution module is divided into two messaging processes, item-to-relationship and relationship-to-item. We define $p_{v_i}^l$ as the representation of item v_i after propagation at layer l, and we initialize the node features of the graph neural network with the embeddings of the items, i.e., $p_{v_i}^0 = e_{v_i}$.

3.1.1 *Item-To-Relationship.*

Following the message propagation paradigm[3], firstly, for different edge types, we construct different messages, and the formula for constructing messages is defined as 2:

$$
m_{v_i,r}^{(l)} = \frac{1}{|\sum_{v_j \in \mathcal{N}_r^{\mathcal{G}_g}(v_i)} w_{ij}^r|} \sum_{v_j \in \mathcal{N}_r^{\mathcal{G}_g}(v_i)} w_{ij}^r p_{v_j}^{(l-1)} \tag{2}
$$

where $m_{v_i,r}^{(l)}$ denotes the message aggregated by node v_i under edge type r at layer l, $\mathcal{N}_r^{\mathcal{G}_g}$ denotes the in-neighbors of node v_i with edge type r in the global $\mathcal{G}_g$. We have constructed multiple relationship-specific messages for each node through the item-to-relationship messaging process.

3.1.2 *Relationship-To-Item.*

In order to aggregate messages from different edge types, we first use an attention mechanism to obtain the weight coefficients from each edge type and then use a weighted sum method to obtain the representation of the item nodes in the next layer. The formula for relationship-to-item is defined as 3:

$$
\begin{aligned}
\pi(v_i, r) &= a^T LeakyReLU(\mathbf{W}_r \ \$ \ [p_{v_i}^{(l-1)} || m_{v_i,r}^{(l)}]) \\
\alpha_{v_i,r} &= \frac{exp(\pi(v_i, r))}{\sum_{k \in \mathcal{R}_g} exp(\pi(v_i, k))} \\
p_{v_i}^{(l)} &= \sum_{k \in \mathcal{R}_g} \alpha_{v_i,k} m_{v_i,k}^{(l)}
\end{aligned}
\tag{3}
$$

where $\mathbf{W}_r \in \mathbb{R}^{2d \times d}$ is the edge type-specific weight matrix, $ represents matrix multiplication, || denotes concatenation operation, $\alpha_{v_i,r} \in \mathbb{R}$ represents attention weight, $\pi(v_i, r) \in \mathbb{R}$ represents the attention coefficient of node v_i and relationship $r \in \mathcal{R}_g$. Finally, the representation $p_{v_i}^{(l)}$ of l-th graph layer for item v_i is obtained by weighting and summing all types of messages.

The item representation under the global co-occurrence graph is obtained by adding the item representations under the convolution of the L-layer graph, i.e., $h_{v_i}^{\mathcal{G}_g} = \sum_{l=0}^{L} p_{v_i}^{(l-1)}$.

3.2 Local Graph Convolution Module

In this section, we propose a local graph convolution module on the local transition graph , which injects user-specific item transition patterns into item representation learning by performing graph convolution operations. Similarly, we define $q_{v_i}^l$ as the representation of item v_i after propagation in the local graph at layer l and then initialize the 0-th layer of the graph neural network with item embedding, i.e., $q_{v_i}^0 = e_{v_i}$. To distinguish the importance of different neighbors, we adopt the attention mechanism, as shown in formula 4:

$$
q_{v_i}^{(l)} = q_{v_i}^{(l-1)} + \sum_{v_j \in \mathcal{N}_{v_i}^{\mathcal{G}_s}} \mathtt{attn}(v_i, v_j) q_{v_j}^{(l-1)} \tag{4}
$$

where $\mathcal{N}_r^{\mathcal{G}_s}$ denotes the in-neighbors of node v_i with edge type $r \in \mathcal{R}_s$, $\mathtt{attn}(v_i, v_j)$ denotes the attention weight between item v_i and v_j. For each item, the importance of its neighbors depends on two factors, semantic similarity and behavior type. Therefore, we propose a new attention mechanism, behavior-aware attention

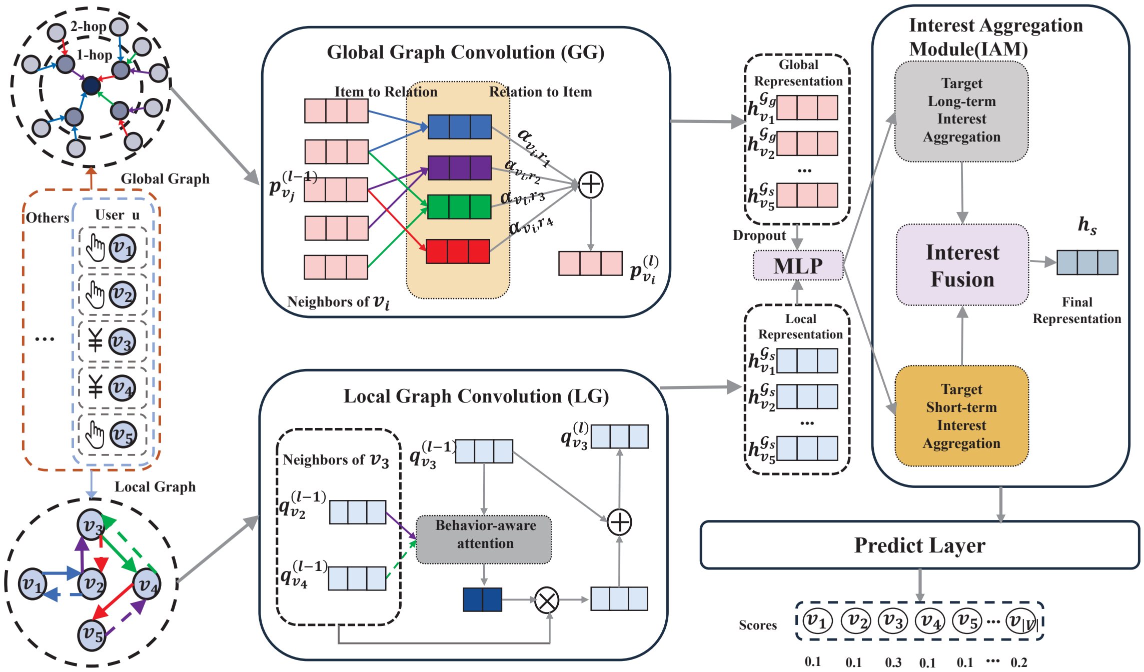

Figure 2: The overall framework of our proposed GHTID.

mechanism.

Behavior-aware attention attention captures the similarity between items of different behaviors to obtain the attention coefficients of different neighboring items. There are different relationships between items for different behaviors. We use different ways of calculating semantic similarity between items for different behavior pairs. The behavior-aware attention coefficients between different items are calculated as formula 5:

$$
\begin{aligned}
\pi(v_i, v_j) &= a_{r_{ij}}^T \mathtt{LeakeyReLU}([\mathbf{W}_1 q_{v_i}^{l-1} || \mathbf{W}_2 q_{v_j}^{l-1}]) \\
\mathtt{attn}(v_i, v_j) &= \frac{exp(\pi(v_i, v_j))}{\sum\limits_{v_k \in \mathcal{N}_{v_i}^{\mathcal{G}_s}} exp(\pi(v_i, v_k))}
\end{aligned} \tag{5}
$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}$ are the projection matrices of the source and target nodes, $r_{ij} \in \mathcal{R}_s$ denotes the relationship between nodes v_i and v_j, $a_{r_{ij}} \in \mathbb{R}^{2d}$ is a relation-specific weight vector, and $\pi(v_i, v_j)$ is the attention coefficients of nodes v_i and v_j. Then we use the softmax function to normalize the attention coefficients of neighbours connecting node v_i to obtain the attention weight of neighbour v_j. The final representation of an item in the on the local transition graph is the item representation of the last layer of the graph network, i.e., $h_{v_i}^{\mathcal{G}_s} = q_{v_i}^{(L)}$.

3.3 Interest Aggregation Module

This module consists of three small modules, target short-term interest aggregation, target long-term interest aggregation, and interest fusion. Before performing interest aggregation, each item v_i obtains a global level representation $h_{v_i}^{\mathcal{G}_g}$ and a local level representation $h_{v_i}^{\mathcal{G}_s}$. The global level representation captures co-occurrence neighbor features under different relationships. The local level representation reflects the personalized preference characteristics of users. We utilize an MLP layer to fuse the representations of two graph neural networks, as shown in Equation 6.

$$
h_{v_i} = \mathtt{Relu}(\mathbf{W}_3[\mathtt{Dropout}(h_{v_i}^{\mathcal{G}_g}) || h_{v_i}^{\mathcal{G}_s}]) \tag{6}
$$

Note that the global graph is constructed from all sequences in the training set. We use the dropout[19] operation for the global graph representation to prevent the model from overfitting.

To capture the user's interest in the target behavior, we introduce the target behavior mask $M = (m_1, m_2, ..., m_i, ..., m_n)$. For each item v_i, if its corresponding behavior b_i is the target behavior, then m_i is 1, otherwise m_i is 0. By aggregating interests related to user's target behavior, interference from auxiliary behavior can be effectively avoided.

3.3.1 *Target Short-Term Interest Aggregation.*

The user's last item is used as a representation of short-term interests, even though its behavior type may not be the target behavior. To reduce the interference of auxiliary behaviors on short-term interests, we use the representation of the last item as a query to aggregate items of the target behavior type in the user's interaction history. Specifically, given the representation of each item in the sequence $(h_{v_1}, h_{v_2}, ..., h_{v_n})$, the sequence representation under short-term interest is as 7.

$$
\begin{aligned}
q_{short} &= h_{v_n} \\
\alpha_i &= a_{short}^T \sigma(\mathbf{W}_4 q_{short} + \mathbf{W}_5 h_{v_i} + b_1) \\
h_s^{short} &= \sum_{i=1}^{n} m_i \alpha_i h_{v_i}
\end{aligned} \tag{7}
$$

where q_{short} is the query of short-term interest, $\mathbf{W}_4, \mathbf{W}_5 \in \mathbb{R}^{d\times d}$ are projection matrices corresponding to the query and key, b_1 is the bias vector, $\sigma(.)$ is the sigmoid function, a_{short} is the attention weight vector of short-term interest, and h_s^{short} is the sequence representation aggregated by short-term interest.

3.3.2 Target Long-Term Interest Aggregation. Long-term interests are related to all items with the target behavior type in the past. Therefore, we use the average representation of all items with the target behavior type in the past as a query to aggregate all items in the user's interaction history. Specifically, given the representation of each item in the sequence (h_{v_1},h_{v_2},...,h_{v_n}), the sequence representation under long-term interest is as 8.

$$\begin{aligned} q_{long} &= \frac{1}{n}\sum_{i=1}^{n} m_i h_{v_i} \\ \beta_i &= a_{long}^T \sigma(\mathbf{W}_6 q_{long} + \mathbf{W}_7 h_{v_i} + b_2) \\ h_s^{long} &= \sum_{i=1}^{n} \beta_i h_{v_i} \end{aligned} \tag{8}$$

Different from short-term interest, the query from long-term interest is obtained by applying an average pooling operation on the item representations with target behavior from a sequence, reflecting the user's long-term preference.

3.3.3 Interest Fusion. The contribution of items in the sequence to prediction varies under different interests. Long-term interest represents the user's general preferences, such as commonly used brands and favorite colors; short-term interest represents the user's current intention, such as travel preparation and birthday gifts. We propose an MLP to aggregate long-term and short-term interests as the ultimate interests of users. The fusion method of the two sequence representations is as 9.

$$h_s = \mathtt{LeakeyReLU}(\mathbf{W}_8[h_s^{short}||h_s^{long}]) \tag{9}$$

Where h_s is the ultimate interests for each user, and $\mathbf{W}_8 \in \mathbb{R}^{d\times 2d}$ is a projection matrice.

3.4 Predicting And Training

In the scenario of multi-behavior recommendation, due to the diversity of behaviors, it is necessary to customize the prediction module to estimate the probability of interaction with the target item under a specific behavior. Consistent with past methods[1, 2], we use different behavior vector to distinguish interaction types and make predictions. Specifically, for sequence s, if the behavior of next interaction is b, then the score of user interacting with item v_i under behavior type b is $y_{s,i}^b = {h_b}^T(h_s \odot e_{v_i})$, where $\odot$ is the element-wise product, h_s is the representation of the sequence, and h_b represents the vector of interaction types. We adopt the cross entropy loss on all items as 10:

$$Loss = -\frac{1}{|O|}\sum_{(s,i,b)\in O} log \frac{exp(y_{s,i}^b)}{\sum_{j=1}^{N} exp(y_{s,j}^b)} \tag{10}$$

Where O is a training set consisting of an interaction sequence s, a positive item v_i, and a target behavior b.

Table 1: Statistics of the datasets after preprocessing.

Dataset	#Users	#items	Behavior Type	#Interaction
ML1M	5645	2357	Exam Like	628,892 223,305
UB	20858	30793	Page View Cart Favorite Purchase	470,731 85,910 28,242 136,250
Rec15	36917	9621	Click Purchase	446,442 233,263
Tmall	17209	16177	Page View Favorite Purchase	831,117 240,901 121,168

4 EXPERIMENTS

To evaluate 's performance, we conducts experiments on several real-world datasets by answering the following research questions:

- **RQ1**: How does our GHTID perform as compared to various state-of-the-art MBSR methods with different settings?
- **RQ2**: How effective are the key modules (e.g. global graph convolution module,local graph convolution module, Interest Aggregation Module) in GHTID?
- **RQ3**: How robust is GHTID to counteract interference from auxiliary behaviors?
- **RQ4**: How do the integration of different types of behavior patterns affect the prediction of target behavior?

4.1 Experiment Settings

4.1.1 Datasets. We used four real datasets, MovieLens 1M(ML1M)[1], UserBehaviors(UB)[2], Tmall[3], and Rec15[4] to evaluate the performance of our model. These four datasets all have different types of behavior, as detailed in the table 1. We adopt the same data processing method as [2]. First, for each behavior, we remove the duplicate items for each user and only keep the earliest ones. Second, we remove the items and users with less than 5 target behavior interactions. Consistent with [2], we only used two type interactions ('Page View' and 'Purchase') in the Tmall and Rec15 datasets. We will conduct an ablation experiment on the behavior types in the third part of the experiment to verify the effectiveness of multi-behavior.

4.1.2 Compared Methods. To evaluate the performance, we compare GHTID with state-of-the-art methods. These models are classified into two types, single-behavior sequential recommendation models and multi-behavior recommendation models, depending on whether the models model behavior types.

Single-Behavior Sequential Recommendation: For a fair comparison, we treat items of different behavior types as the same behavior in the training phase and evaluate them under the target behavior in the validation and testing phases.

[1] https://grouplens.org/datasets/movielens/1m
[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
[3] https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
[4] https://recsys.acm.org/recsys15/challenge

- **FPMC** [18]. A joint matrix decomposition and first-order Markov approach that models user preferences as well as first-order transfer information of items, respectively.
- **TransRec** [5]. A translation-based approach models the "third-order relationship" between the user, the item, and the next item.
- **SASRec** [8]. It encodes the item sequence by the transformer and autoregressively makes the next item prediction.
- **TiSASRec** [12]. An improvement work based on SASRec incorporates the time interval relationship between items into the Self-Attention module.
- **GCEGNN** [24]. A graph neural network-based approach that learns item representations in the local and global graphs, respectively, and fuses reverse positional encoding to obtain session representations for prediction.

Multi-Behavior Sequential Recommendation: MBSR methods pay extra attention to the behavior sequence of items.

- **TransRec++** [32]. It enhances TransRec to model item transition relationships as different vectors based on behavior type
- **DMT** [4]. It divides the item sequence into behavior-specific item subsequences based on behavior type and models each behavior-specific subsequence separately.
- **MGNN-Spred** [23]. It learns sequence-level behavior dependencies by constructing behavior-specific item transformation graphs.
- **M-SR** [17]. It uses GGNN and GRU to encode item sequences and behavior sequences, respectively, and note that we remove the knowledge embedding module from the original model
- **MBSTR** [31]. It employs a pairwise behavior-specific relative position encoding to improve Self-Attention. In the prediction phase, it treats the item prediction task with multiple behavior types as multi-task learning with the latest multi-task learning CGC module configured.
- **GPG4HSR** [2]. It construct global and local graphs based on adjacent item transitions, where the global graph learns pairwise behavior-specific relationships and the local graph injects behavior context to the item representation.

4.1.3 Evaluation Protocol. We adopt the "leave one out" evaluation strategy, as is commonly used in most Sequential Recommendation work[8, 28]. We adopt performance evaluation metrics widely used in Top-N recommendations, including the normalized discounted cumulative gain (NDCG@N) and the hit ratio (HR@N). In this experiment, we set N=10. Due to the inconsistent with non-sampling methods[9, 10], we adopt a non-sampling evaluation approach, i.e., scoring items in the item set under the target behavior for evaluation purposes.

4.1.4 Implementation Details. We implement GHTID with Pytorch and fine-tune the hyperparameters on the validation set. For the baseline models, we used the version provided by the authors. Following previous work[2], the embedding size is 64, the sequence length is 50, the batch size is 128, the optimizer is Adam, and the learning rate is 0.001. We set a 2-layer Transformer encoder and single-headed attention for all Transformer-based models. For all hyperparameters, we tuned the parameters on the validation set according to the settings in the original paper.

4.2 Performance Comparison (RQ1)

We report a detailed performance comparison of the different methods in Table 2 and summarize the observations as follows:

GHTID outperforms all the baseline models on the four datasets. The average improvement of $HR@10$ and $NDCG@10$ over the best baseline model on the ML-1M dataset are 13.90% and 17.90%, 23.34% and 32.68% on the UB dataset, 19.86% and 18.07% on the Rec15 dataset, 18.11% and 21.17% on the Tmall dataset, which proves the effectiveness of GHTID. The performance improvement can be attributed to: *i*) GHTID considers heterogeneous item transitions across types from global and local perspectives and can learn item-level behavior dependencies and effectively obtain better item representations. *ii*) Compared with other models, GHTID aggregats interest from target behavior and reduces the interference of auxiliary behavior.

Modeling behavior sequences can help improve the performance of single-behavior sequential recommendation models. Some multi-behavior sequential recommendation methods (e.g. DMT) perform even weaker than single-behavior sequential recommendation methods (e.g. SASRec) due to the great differences in the design of different models. However, the overall trend is that multi-behavior sequential recommendation methods outperform single-behavior sequential recommendation methods of similar architectures. The three groups of models, TransRec and TransRec++, SASRec and MBSTR, and GCE-GNN and GPG4HSR, have similar underlying architectures, and the latter are all improved versions of the former after modeling multi-behavior sequences. Comparing the performance of these three models, we can find that the latter has some performance improvement compared with the former.

Item-level behavior dependency modeling approaches generally outperform sequence-level behavior dependency modeling approaches. Item-level behavior dependency modeling methods MBSTR, GPG4HSR, and GHTID generally outperform other multi-behavior sequential recommendation methods, especially relative to sequence-level behavior dependency modeling methods DMT, MGNN-SPred. Sequence-level behavior dependency modeling methods are sometimes inferior to single-behavior sequential recommendation methods. This performance difference can be attributed to sequence-level behavior dependency modeling approaches ignoring the correlation between items across behavior types.

4.3 Ablation Study (RQ2)

In this section, we consider three key components, the global graph convolution module(GG), the local graph convolution module(LG), and the interest aggregation module(IAM). To study the effectiveness of each component, we propose the following variants of GHTID.

- **GHTID w/o GG:** The global graph convolution module is removed so that the representation of the items takes into account only the information in the local sequence.
- **GHTID w/o LG:** The local graph convolution module is removed so that the item representation considers only the item co-occurrence information at the global level.
- **GHTID w/o IAM:** The interest aggregation module is removed, and as an alternative, we use the last item representation of the sequence as the representation of the sequence.

Table 2: Overall model performance on four datasets, with the metrics of HR@N and NDCG@N (N=10). The best score and the second best score in each row are bolded and underlined, respectively. Improvements over the best baseline method are indicated in the last row.

Methods	ML1M		UB		Rec15		Tmall	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
FPMC	0.1086	0.0510	0.0467	0.0249	0.3829	0.2102	0.0352	0.0191
TransRec	0.0852	0.0409	0.0589	0.0342	0.3697	0.1928	0.0374	0.0325
SASRec	0.1350	0.0651	0.0744	0.0412	0.3615	0.1889	0.0862	0.0521
TiSASRec	0.1327	0.0638	0.0736	0.0415	0.4093	0.2056	0.0746	0.0445
GCEGNN	0.1319	0.0618	0.0611	0.0328	0.4026	0.2053	0.0823	0.0487
TransRec++	0.1088	0.0508	0.0661	0.0413	0.4064	0.2209	0.0593	0.0377
DMT	0.1158	0.0521	0.0613	0.0343	0.3942	0.2036	0.0642	0.0335
MGNN-Spred	0.1134	0.0548	0.0727	0.0386	0.4164	0.2108	0.0449	0.0236
M-SR	0.1349	0.0647	0.0784	0.0407	0.4315	0.2327	0.0811	0.0498
MBSTR	0.1431	0.0716	0.0904	0.0453	0.4239	0.2274	0.0905	0.0516
GPG4HSR	0.1460	0.0737	0.0830	0.0462	0.4198	0.2160	0.0944	0.0548
GHTID	**0.1663**	**0.0869**	**0.1124**	**0.0613**	**0.5081**	**0.2685**	**0.1115**	**0.0664**
Improv.	13.90%	17.90%	23.34%	32.68%	19.86%	18.07%	18.11%	21.17%

Table 3: Ablation study of GHTID

Methods	UB		Tmall	
	HR@10	NDCG@10	HR@10	NDCG@10
GHTID w/o GG	0.1042	0.0541	0.1003	0.0528
GHTID w/o LG	0.0917	0.0498	0.0947	0.0512
GHTID w/o IAM	0.1057	0.0569	0.1038	0.0557
GHTID	**0.1124**	**0.0613**	**0.1115**	**0.0664**
SASRec	0.0744	0.0412	0.0862	0.0521
SASRec+IAM	**0.0847**	**0.0512**	**0.0922**	**0.0535**
MBSTR	0.0904	0.0453	0.0905	0.0516
MBSTR+IAM	**0.0961**	**0.0523**	**0.0967**	**0.0551**

Table 3 reports the performance comparison of GHTID and its three variants on UB and Tmall datasets (HR@10, NDCG@10). In addition, IAM can be added to any sequence encoder to reduce the interference of auxiliary behavior. We add variants based on SASRec and MBSTR, respectively.

Comparing the predicted performance of GHTID and its three variants, significant performance decreases occur when removing any key component. In particular, the performance gap between the GHTID and GHTID w/o LG variants shows the advantage of learning heterogeneous item transitions for each user. The gap in performance between GHTID and GHTID w/o GG indicates that GG effectively captures item co-occurrence information from other sequences. The performance gap between GHTID and GHTID w/o IAM shows the superiority of the IAM in aggregating interests from target behavior. Furthermore, the performance gap between SASRec, MBSTR, and its variants shows that the IAM can bring performance improvements to the sequence encoder, which indicates the effectiveness of IAM.

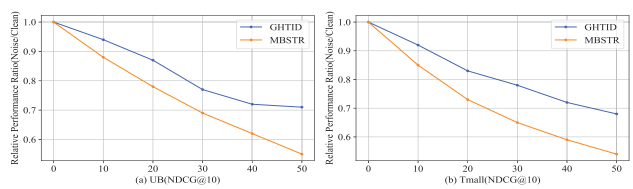

Figure 3: Relative performance drop on dataset UB and Tmall when the test data are corrupted by synthetic noises on auxiliary behavior. The x-axis is the percentage of corrupted data from auxiliary behavior. The y-axis is the ratio of the performance with noisy test data to the performance with clean training data.

4.4 Robustness to Auxiliary Behavior(RQ3)

We conduct robustness experiments to analyze GHTID's ability to resist interference from auxiliary behaviors. Specifically, we added noise to the auxiliary behaviors in the test data at a specific ratio. The way noise is added to each data comes from a random one of Mask, Crop, Reorder[27], Substitute, and Insert[15]. The proportion of disrupted data ranges from 0 to 50 and increases in increments of 10 per cent. We evaluate the percentage decrease in performance on NDCG@10 for the UB and Tmall datasets, as shown in the Figure 3.

We can see that compared to MBSTR, GHTID's performance decline is significantly slower. Moreover, the larger the noise ratio, the more pronounced the performance gap. This indicates that GHTID, which can adaptively extract signals related to the target behavior from auxiliary behaviors, does help improve robustness to auxiliary behavior.

4.5 Effects of Behavior types(RQ4)

To verify the effectiveness of multi-behavior, we conduct a data ablation study in this section to evaluate the model performance after incorporating different combinations of auxiliary behaviors.

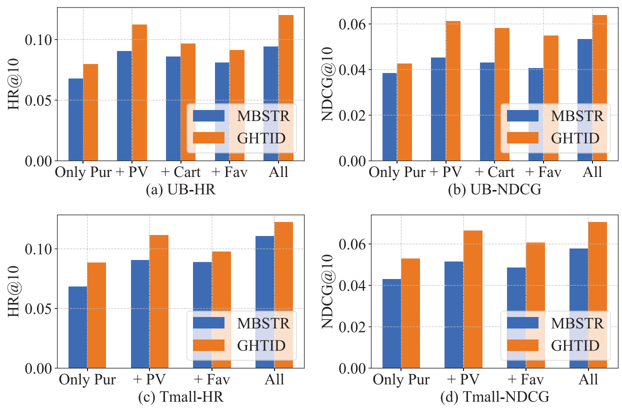

Figure 4: Effect of auxiliary behaviors.

We chose two multi-behavior datasets (number of behavior types greater than 2), UB and Tmall. For brevity, we simplify the behavior types as Page View(PV), add to Cart (Cart), Favorite(Fav), and Purchase(Pur). We designed three types of variants. *i*)**Only Pur**: contains only the target behavior. *ii*)**+ ***(* denotes an auxiliary behavior): includes the target behavior and some auxiliary behavior. *iii*)**All**:includes the target behavior and all auxiliary behaviors. We evaluated our performance on two models, MBSTR and GHTID, and the evaluation results are shown in Figure 4.

The experimental results show that adding any of the other auxiliary behaviors can improve the prediction performance of the target behavior compared to the variant using only the target behavior. The performance improvement of '**+ PV**' compared to the other variants with only one auxiliary behavior indicates the importance of Page View behavior for predicting Purchase behavior. The best performance is obtained for '**All**', emphasizing the necessity of integrating more auxiliary behaviors under the target behavior to help with recommendations.

5 RELATED WORK

Sequential recommendation(SR) mainly models the item transitions in a sequence to capture the current and recent user preferences. Based on the number of behavior types used, we divide the work of SR into two categories: **single-behavior sequential recommendation** (SBSR) and **multi-behavior sequential recommendation** (MBSR).

5.1 Single-Behavior Sequential Recommendation

Most early works on SBSR used Markov chain to model item sequences and predicted the next item of the user by calculating the transition probability matrix[6]. FPMC[18] combines matrix factorization to capture user's long-term preference and first-order Markov chain to model sequential pattern. Since Recurrent Neural Network (RNN) has a natural ability to model sequences, it was the first to be applied in the field of SR[7, 11]. Convolutional Neural Network (CNN) has also been explored for its potential to apply to SR, such as Caser[20], which adopts customized filters to learn the collective dependencies in the sequence. In recent years, transformer and Graph Neural Network (GNN) have been widely applied to SR and achieved excellent performance. SASRec[8] adopts a left-to-right transformer for autoregressive item prediction. SR-GNN[25] designs item transition graphs for each item sequence and performs item and sequence representation learning with Gated Graph Neural Network (GGNN)[13]. SBSR methods only consider one type of behavior, ignoring the correlation signals between different types of interactions.

5.2 Multi-Behavior Sequential Recommendation

In recent years, MBSR has received more and more attention. Because modeling multi-behavior item sequences can better reflect users' interests and behavior patterns and improve the accuracy of recommendations. According to the modeling methods of behavior sequences, we divided them into three categories.

The first category is to model the behavior sequence as auxiliary information. These methods either directly aggregate the behavior embeddings and item embeddings as the input of the model[33], or separately model the item sequence and behavior sequence and then aggregate them at the end[14, 17, 21]. The second category is sequence-level behavior dependency modeling. This type of method first divides the original item sequence into different sub-sequences according to different types of behavior, then models each sub-sequence separately, and finally aggregates them[4, 16, 29]. This method ignores the cross-type transitions between items because it separates the item sequences of different behaviors.

The third type is item-level behavior dependency modeling. This type of methods build relationships between items based on behavior, and mine the item transitions within the same behavior and across behaviors. However, the main drawback of existing methods is the lack of exploration of heterogeneous item transitions. Some approaches[26, 31] focus only on individual user interactions, ignoring global heterogeneous item transitions. Other approaches[2] do not explicitly model heterogeneous item transitions in local item representation learning. Different from the above works, our proposed GHTID explores the item-level behavior dependency modeling from global and local perspectives.

6 CONCLUSION

In this work, we propose a multi-behavior sequential recommendation framework GHTID. It addresses two major challenges in multi-behavior sequential recommendation systems: heterogeneous item transitions and interference from auxiliary behaviors. Firstly, we propose a graph convolution module based on the global item-item co-occurrence graph and a graph convolution module based on the local item-item transition graph. They learn heterogeneous item transition relationships at the global and local levels. In addition, our proposed interest aggregation module aggregates signals from the target behavior into long-term and short-term interests, effectively reducing interference from auxiliary behaviors. Extensive experiments on four public datasets validate the effectiveness of GHTID.

In future work, we will consider user profiling and item attributes to expand our framework.

REFERENCES

[1] Chong Chen, Min Zhang, Yongfeng Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 19–26.

[2] Weixin Chen, Mingkai He, Yongxin Ni, Weike Pan, Li Chen, and Zhong Ming. 2022. Global and Personalized Graphs for Heterogeneous Sequential Recommendation by Learning Behavior Transitions and User Intentions. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 268–277.

[3] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. PMLR, 1263–1272.

[4] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. 2020. Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2493–2500.

[5] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 161–169.

[6] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.

[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] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 1748–1757.

[10] Dong Li, Ruoming Jin, Jing Gao, and Zhi Liu. 2020. On sampling top-k recommendation evaluation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2114–2124.

[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] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*. 322–330.

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

[14] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1734–1743.

[15] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479* (2021).

[16] Jinwei Luo, Mingkai He, Xiaolin Lin, Weike Pan, and Zhong Ming. 2022. Dual-Task Learning for Multi-Behavior Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1379–1388.

[17] Wenjing Meng, Deqing Yang, and Yanghua Xiao. 2020. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in Information Retrieval*. 1091–1100.

[18] 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*. 811–820.

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[20] 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*. 565–573.

[21] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. 2020. Attentive sequential models of latent intent for next item recommendation. In *Proceedings of The Web Conference 2020*. 2528–2534.

[22] Yu Tian, Jianxin Chang, Yanan Niu, Yang Song, and Chenliang Li. 2022. When Multi-Level Meets Multi-Interest: A Multi-Grained Neural Model for Sequential Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1632–1641.

[23] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *Proceedings of the web conference 2020*. 3056–3062.

[24] 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.

[25] 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.

[26] Lianghao Xia, Chao Huang, Yong Xu, and Jian Pei. 2022. Multi-Behavior Sequential Recommendation with Temporal Graph Transformer. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[27] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.

[28] Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled side information fusion for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1611–1621.

[29] Yuhao Yang, Chao Huang, Lianghao Xia, Yuxuan Liang, Yanwei Yu, and Chenliang Li. 2022. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2263–2274.

[30] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 7370–7377.

[31] Enming Yuan, Wei Guo, Zhicheng He, Huifeng Guo, Chengkai Liu, and Ruiming Tang. 2022. Multi-behavior sequential transformer recommender. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1642–1652.

[32] Zhuoxin Zhan, Mingkai He, Weike Pan, and Zhong Ming. 2022. TransRec++: Translation-based sequential recommendation with heterogeneous feedback. *Frontiers Comput. Sci.* 16, 2 (2022), 162615.

[33] Meizi Zhou, Zhuoye Ding, Jiliang Tang, and Dawei Yin. 2018. Micro behaviors: A new perspective in e-commerce recommender systems. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 727–735.