Multi-Behavior Hypergraph-Enhanced Transformer for Sequential Recommendation

Yuhao Yang
University of Hong Kong
Hong Kong, China
yuhao-yang@outlook.com

Chao Huang*
University of Hong Kong
Hong Kong, China
chaohuang75@gmail.com

Lianghao Xia
University of Hong Kong
Hong Kong, China
aka_xia@foxmail.com

Yuxuan Liang
National University of Singapore
Singapore, Singapore
yuxliang@outlook.com

Yanwei Yu
Ocean University of China
Qingdao, China
yuyanwei@ouc.edu.cn

Chenliang Li
Wuhan University
Wuhan, China
cllee@whu.edu.cn

ABSTRACT

Learning dynamic user preference has become an increasingly important component for many online platforms (*e.g.,* video-sharing sites, e-commerce systems) to make sequential recommendations. Previous works have made many efforts to model item-item transitions over user interaction sequences, based on various architectures, *e.g.,* recurrent neural networks and self-attention mechanism. Recently emerged graph neural networks also serve as useful backbone models to capture item dependencies in sequential recommendation scenarios. Despite their effectiveness, existing methods have far focused on item sequence representation with singular type of interactions, and thus are limited to capture dynamic heterogeneous relational structures between users and items (*e.g.,* page view, add-to-favorite, purchase). To tackle this challenge, we design a <u>M</u>ulti-<u>B</u>ehavior <u>H</u>ypergraph-enhanced <u>T</u>ransformer framework (MBHT) to capture both short-term and long-term cross-type behavior dependencies. Specifically, a multi-scale Transformer is equipped with low-rank self-attention to jointly encode behavior-aware sequential patterns from fine-grained and coarse-grained levels. Additionally, we incorporate the global multi-behavior dependency into the hypergraph neural architecture to capture the hierarchical long-range item correlations in a customized manner. Experimental results demonstrate the superiority of our MBHT over various state-of-the-art recommendation solutions across different settings. Further ablation studies validate the effectiveness of our model design and benefits of the new MBHT framework. Our implementation code is released at: https://github.com/yuh-yang/MBHT-KDD22.

CCS CONCEPTS

• **Information systems → Recommender systems**.

---

*Chao Huang is the corresponding author.

KEYWORDS

Sequential Recommendation, Graph Neural Networks, Hypergraph Learning, Multi-Behavior Recommendation

1 INTRODUCTION

Recommendation models have emerged as the core components of many online applications [34], such as social media platforms [36], video streaming services [10] and online retail systems [24]. Due to the highly practical value of sequential behavior modeling in various online platforms, sequential recommendation has been widely adopted in online platforms, with the aim of forecasting future users' interacted item based on their past behavior sequences [1, 19].

Generally, in the sequential recommendation scenario, the systems rely on the item sequences to model time-evolving user preferences. Upon this learning paradigm, many sequential recommender systems have been proposed to encode the item dependencies based on various neural techniques and provided stronger performance, *e.g.,* recurrent neural recommendation architecture-GRU4Rec [7] and convolution-based sequence encoder model-Caser [20]. Inspired by the Transformer framework, self-attention has been utilized to capture the item-item pairwise correlations in SASRec [13] and BERT4Rec [19]. Recently, graph neural networks (GNNs) have shown effectiveness in sequential recommender systems. Owing to strength of graph convolutions, GNN-based methods (*e.g.,* SR-GNN [25], GCSAN [28]) are developed to learn item transitions with message passing over the constructed item graph structures.

Despite their effectiveness, most of existing works have thus far focused on sequential behavior pattern encoding for singular type of interactions (*e.g.,* clicks or purchases), without taking multi-typed user-item relationships into consideration. However, in practical online platforms, user behaviors often exhibit both time-dependent and multi-typed, involving different types of user-item interactions, *e.g.,,* page view, add-to-favorite, purchase. As illustrated in Figure 1, customers may click and add their interested products to their favorite lists before purchasing them. Additionally, we also visualize
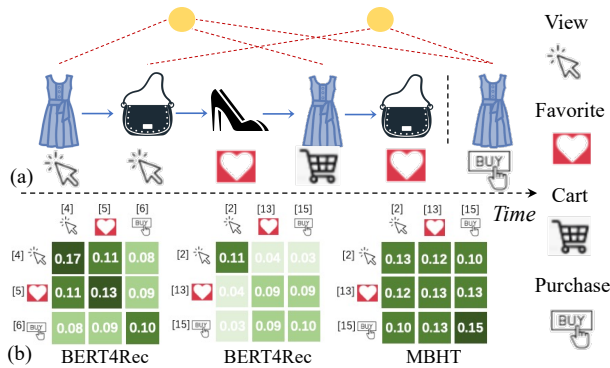
Figure 1: (a) Illustration example of sequential recommendation with multi-behavior dynamics. (b) Learned behavior-aware dependency weights for short-term item correlations among neighboring {[4], [5], [6]} **and long-range dependencies among** {[2], [13], [15]} **by BERT4Rec and our MBHT.**

the learned dependency weights by the baseline BERT4Rec and our MBHT method to capture behavior-aware short-term item correlations (among neighboring items {[4], [5], [6]}) and long-term item-wise dependencies (among {[2], [13], [15]}). We can observe that the global multi-behavior dependencies can be better distilled by our MBHT as compared to BERT4Rec. Hence, effectively enhancing the user preference learning with the exploration of heterogeneous user-item interactions in a dynamic environment, is also the key to making accurate sequential recommendations. Nevertheless, this task is not trivial due to the following challenges:

- **Dynamic Behavior-aware Item Transitions**. How to explicitly capture the dynamic behavior-aware item transitions with multi-scale temporal dynamics remains a challenge. There exist different periodic behavior patterns (*e.g.,* daily, weekly, monthly) for different categories of items (*e.g.,* daily necessities, seasonal clothing) [9, 17]. Therefore, it is necessary to explicitly capture multi-scale sequential effects of behavior-aware item transitional patterns from fine-grained to coarse-grained temporal levels.

- **Personalized Global Multi-Behavior Dependencies**. The implicit dependencies across different types of behaviors over time vary from user to user. For example, due to the personalized and diverse user interaction preferences, some people would like to add products to their favorite list if they show interested in the items. Others may prefer to generate their favorite item list with products they are very likely to buy. That is to say, for different users, multi-behaviors have various time-aware dependencies on their interests. Moreover, multi-behavior item-wise dependencies are beyond pairwise relations and may exhibit triadic or event higher-order. Hence, the designed model requires a tailored modeling of diverse users' multi-behavior dependencies with a dynamic multi-order relation learning paradigm.

Contribution. This work proposes a <u>M</u>ulti-<u>B</u>ehavior <u>H</u>ypergraph-enhanced <u>T</u>ransformer (MBHT) to capture dynamic item dependencies with behavior type awareness. Our developed MBHT framework consitens of two key learning paradigms to address the aforementioned challenges correspondingly. (1) **Behavior-aware Sequential Patterns**. We propose a multi-scale Transformer module to comprehensively encode the multi-grained sequential patterns

from fine-grained level to coarse-grained level for behavior-aware item transitions. To improve the efficiency of our sequential pattern encoder, we equip our multi-scale Transformer with the self-attentive projection based on low-rank factorization. To aggregate scale-specific temporal effects, a multi-scale behavior-aware pattern fusion is introduced to integrate multi-grained item transitional signals into a common latent representation space. (2) **Global Modeling of Diverse Multi-Behavior Dependencies**. We generalize the modeling of global and time-dependent cross-type behavior dependencies with a multi-behavior hypergraph learning paradigm. We construct the item hypergraph structures by unifying the latent item-wise semantic relateness and item-specific multi-behavior correlations. Technically, to capture the global item semantics, we design the item semantic dependence encoder with metric learning. Upon the hypergraph structures, we design the multi-behavior hyperedge-based message passing schema for refining item embeddings, which encourages the long-range dependency learning of different types of user-item relationships. Empirically, MBHT is able to provide better performance than state-of-the-art methods, *e.g.,* BERT4Rec [19], HyperRec [21], SURGE [1], MB-GMN [27].

The main contributions are summarized as follows:

- This work proposes a new framework named MBHT for sequential recommendation, which uncovers the underlying dynamic and multi-behavior user-item interaction patterns.

- To model multi-grained item transitions with the behavior type awareness, we design a multi-scale Transformer which is empowered with the low-rank and multi-scale self-attention projection, to maintain the evolving relation-aware user interaction patterns.

- In addition, to capture the diverse and long-range multi-behavior item dependencies, we propose a multi-behavior hypergraph learning paradigm to distill the item-specific multi-behavior correlations with global and customized sequential context injection.

- We perform extensive experiments on three publicly available datasets, to validate the superiority of our proposed MBHT over various state-of-the-art recommender systems. Model ablation and case studies further show the benefits of our model.

2 PROBLEM FORMULATION

In this section, we introduce the primary knowledge and formulates the task of multi-behavior sequential recommendation.

Behavior-aware Interaction Sequence. Suppose we have a sequential recommender system with a set of I users $u_i \in \mathcal{U}$ where $|\mathcal{U}| = I$. For an individual user u_i, we define the behavior-aware interaction sequence $S_i = [(v_{i,1}, b_{i,1}), ..., (v_{i,j}, b_{i,j}), ..., (v_{i,J}, b_{i,J})]$ with the consideration of item-specific interaction type, where J denotes the length of temporally-ordered item sequence. Here, we define $b_{i,j}$ to represent the behavior type of the interaction between user u_i and j-th item v_j in S_i, such as page view, add-to-favorite, add-to-cart and purchase in e-commerce platforms.

Task Formulation. In our multi-behavior sequential recommender system, different types of interaction behaviors are partitioned into *target behaviors* and *auxiliary behaviors*. Specifically, we regard the interaction with the behavior type we aim to predict as target behaviors. Other types of user behaviors are defined as auxiliary behaviors to provide various behaviour contextual information about users' diverse preference, so as to assist the recommendation

task on the target type of user-item interactions. For example, in many online retail platforms, purchase behaviors can be considered as the prediction targets, due to their highly relevance to the Gross Merchandise Volume (GMV) in online retailing to indicate the total sales value for merchandise [9, 24]. We formally present our studied sequential recommendation problem as follows:

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

- **Output**: The learning function that estimates the probability of user u_i will interact with the item v_{J+1} with the target behavior type at the future $(J + 1)$-th time step.

3 METHODOLOGY

Figure 2 presents the overall architecture of our proposed MBHT model which consists of three key modules: i) Multi-scale modeling of behavior-aware transitional patterns of user preference; ii) Global learning of multi-behavior dependencies of time-aware user interactions; iii) Cross-view aggregation with the encoded representations of sequential behavior-aware transitional patterns and hypergraph-enhanced multi-behavior dependencies.

3.1 Multi-Scale Modeling of Behavior-aware Sequential Patterns

In this section, we present the technical details of our MBHT in capturing the behavior-aware user interest with multi-scale dynamics.

3.1.1 **Behavior-aware Context Embedding Layer**. To inject the behavior-aware interaction context into our sequential learning framework, we design the behavior-aware context embedding layer to jointly encode the individual item information and the corresponding interaction behavior contextual signal. Towards this end, given an item v_j, we offer its behavior-aware latent representation $\mathbf{h}_j \in \mathbb{R}^d$ with the following operation:

$$\mathbf{h}_j = \mathbf{e}_j \oplus \mathbf{p}_j \oplus \mathbf{b}_j \tag{1}$$

where $\mathbf{e}_j \in \mathbb{R}^d$ represents the initialized item embedding. $\mathbf{b}_j \in \mathbb{R}^d$ is the behavior type embedding corresponding to the interaction type (*e.g.*, page view, add-to-favorite) between user u_i and item v_j. Here, $\mathbf{p}_j \in \mathbb{R}^d$ represents the learnable positional embedding of item v_j which differentiates the temporally-ordered positional information of different interacted items. After this context embedding layer, we can obtain the item representation matrix $\mathbf{H} \in \mathbb{R}^{J \times d}$ for the behavior-aware interacted item sequence S_i of user u_i.

3.1.2 **Multi-Scale Transformer Layer**. In practical recommendation scenarios, user-item interaction preference may exhibit multi-scale transitional patterns over time. For instance, users often purchase different categories of products (*e.g.*, daily necessities, clothes, digital devices) with different periodic trends, such as daily or weekly routines [10]. To tackle this challenge, we design a multi-scale sequential preference encoder based on the Transformer architecture to capture the multi-grained behavior dynamics in the behavior-aware interaction sequence S_i of users ($u_i \in \mathcal{U}$).

Low-Rank Self-Attention Module. Transformer has shown its effectiveness in modeling relational data across various domains

(*e.g.*, language [31], vision [15]). In Transformer framework, self-attention serves as the key component to perform the relevance-aware information aggregation among attentive data points (*e.g.*, words, pixels, items). However, the high computational cost (*i.e.*, quadratic time complexity) of the self-attention mechanism limits the model scalability in practical settings [14]. Motivated by the design of Transformer structure in [22], we design a low-rank-based self-attention layer without the quadratic attentive operation, to approximate linear model complexity.

In particular, different from the original scaled dot-product attention for pairwise relation encoding, we generate multiple smaller attention operations to approximate the original attention with low-rank factorization. We first define two trainable projection matrices $\mathbf{E} \in \mathbb{R}^{\frac{J}{C} \times J}$ and $\mathbf{F} \in \mathbb{R}^{\frac{J}{C} \times J}$ to perform the low-rank embedding transformation. Here, C denotes the low-rank scale and $\frac{J}{C}$ represents the number of low-rank latent representation spaces over the input behavior-aware interaction sequence S_i. Formally, we represent our low-rank self-attention as follows:

$$\widehat{\mathbf{H}} = \text{softmax}(\frac{\mathbf{H} \cdot W^Q (\mathbf{E} \cdot \mathbf{H} \cdot W^K)^{\mathsf{T}}}{\sqrt{d}}) \cdot \mathbf{F} \cdot \mathbf{H} \cdot W^V \tag{2}$$

where W^Q, W^K, W^V are learnable transformation matrices for embedding projection. In our low-rank self-attention module, $\mathbf{E}$ and $\mathbf{F}$ are utilized to project the ($\mathbb{R}^{J \times d}$)-dimensional key and value transformed representations $\mathbf{H} \cdot W^K$ and $\mathbf{H} \cdot W^V$ into ($\mathbb{R}^{\frac{J}{C} \times d}$)-dimensional latent low-rank embeddings $\widehat{\mathbf{H}} \in \mathbb{R}^{\frac{J}{C} \times d}$. In summary, with the low-rank factor decomposition over the original attention operations, we calculate the context mapping matrix $\mathbf{M} = \frac{\mathbf{H} \cdot W^Q (\mathbf{H} \cdot W^K)^T}{\sqrt{d}}$ with the dimension of $\mathbb{R}^{J \times \frac{J}{C}}$ as compared to the original dimension $\mathbb{R}^{J \times J}$ in the vanilla self-attention mechanism. By doing so, the computational cost of our behavior sequence encoder can be significantly reduced from the $O(J \times J)$ to $O(J \times \frac{J}{C})$ given that the low-rank projected dimension J/C is often much smaller than J, *i.e.*, $J/C \ll J$.

Multi-Scale Behavior Dynamics. To endow our MBHT model with the effective learning of multi-scale behaviour transitional patterns, we propose to enhance our low-rank-based transformer with a hierarchical structure, so as to capture granularity-specific behavior dynamics. To be specific, we develop a granularity-aware aggregator to generate granularity-specific representation $\mathbf{g}_p$ which preserves the short-term behavior dynamic. Here, we define p as the length of sub-sequence for a certain granularity. We formally present our granularity-aware emebdding generation with the aggregated representation $\mathbf{\Gamma}^p \in \mathbb{R}^{\frac{J}{p} \times d}$ and $\boldsymbol{\gamma} \in \mathbb{R}^d$ as follows:

$$\mathbf{\Gamma}^p = \{\boldsymbol{\gamma}_1, ..., \boldsymbol{\gamma}_{\frac{J}{p}}\} = [\eta(\mathbf{h}_1, ..., \mathbf{h}_p); ...; \eta(\mathbf{h}_{J-p+1}, ..., \mathbf{h}_J)] \tag{3}$$

where $\eta(\cdot)$ represents the aggregator to capture the short-term behavior-aware dynamics. Here, we utilize the mean pooling to perform the embedding aggregation. After that, we feed the granularity-aware behavior representations into a self-attention layer for encoding granularity-specific behavior pattern as shown below:

$$\mathbf{H}^p = \text{softmax}(\frac{\mathbf{\Gamma}^p \cdot W_p^Q (\mathbf{\Gamma}^p \cdot W_p^K)^{\mathsf{T}}}{\sqrt{d}}) \cdot \mathbf{\Gamma}^p \cdot W_p^V \tag{4}$$
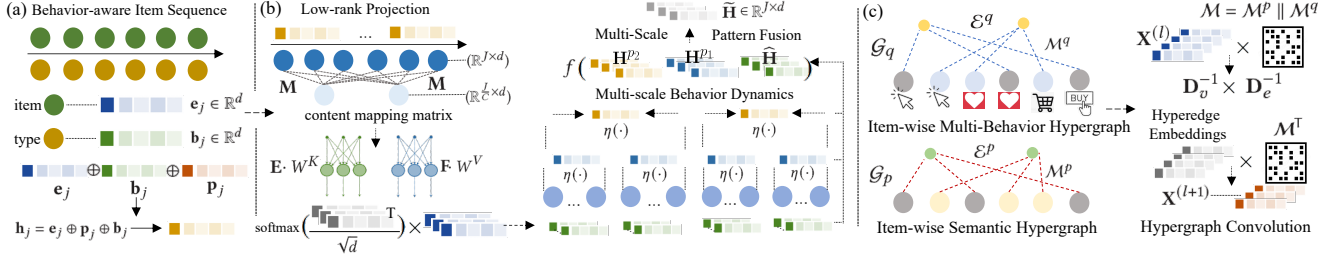
Figure 2: MBHT's model flow. (a) We inject the behavior-aware interaction context into item embeddings $\mathbf{h}_j = \mathbf{e}_j \oplus \mathbf{p}_j \oplus \mathbf{b}_j$. (b) Multi-scale transformer architecture to capture behavior-aware transitional patterns via low-rank self-attention and multi-scale sequence aggregation. Scale-specific behavior patterns are fused through the fusion function $\widetilde{\mathbf{H}} = f(\widehat{\mathbf{H}} \| \mathbf{H}^{p_1} \| \mathbf{H}^{p_2})$. (c) We capture the global and personalized multi-behavior dependency learning with our hypergraph neural architecture over $\mathcal{G}$.

$\mathbf{H}^p \in \mathbb{R}^{\frac{J}{p} \times d}$ encodes the short-term transitional patterns over different item sub-sequences. In our MBHT framework, we design a hierarchical Transformer network with two different scale settings p_1 and p_2. Accordingly, our multi-scale Transformer can produce three scale-specific sequential behavior embeddings $\widehat{\mathbf{H}} \in \mathbb{R}^{J \times d}$, $\mathbf{H}^{p_1} \in \mathbb{R}^{\frac{J}{p_1} \times d}$, $\mathbf{H}^{p_2} \in \mathbb{R}^{\frac{J}{p_2} \times d}$.

3.1.3 **Multi-Scale Behaviour Pattern Fusion.** To integrate the multi-scale dynamic behavior patterns into a common latent representation space, we propose to aggregate the above encoded scale-specific embeddings with a fusion layer presented as follows:

$$\widetilde{\mathbf{H}} = f(\widehat{\mathbf{H}} \| \mathbf{H}^{p_1} \| \mathbf{H}^{p_2}) \tag{5}$$

Here $f(\cdot)$ represents the projection function which transforms $\mathbb{R}^{(\frac{J}{C} + \frac{J}{p_1} + \frac{J}{p_2}) \times d}$ dimensional embeddings into $\mathbb{R}^{J \times d}$ dimensional representations corresponding to different items ($v_j \in S_i$) in the behavior-aware interaction sequence S_i of user u_i. Here, $\|$ denotes the concatenation operation over different embedding vectors.

Multi-Head-Enhanced Representation Spaces. In this part, we propose to endow our behavior-aware item sequence encoder with the capability of jointly attending multi-dimensional interaction semantics. In particular, our multi-head sequential pattern encoder projects the $\mathbf{H}$ into N latent representation spaces and performs head-specific attentive operations in parallel.

$$\widetilde{\mathbf{H}} = (\text{head}_1 \| \text{head}_2 \| \cdots \| \text{head}_N) \mathbf{W}^D \tag{6}$$
$$\text{head}_n = f(\widehat{\mathbf{H}}_n \| \mathbf{H}_n^{p_1} \| \mathbf{H}_n^{p_2})$$

where $\widehat{\mathbf{H}}_n$, $\mathbf{H}_n^{p_1}$ and $\mathbf{H}_n^{p_2}$ are computed with head-specific projection matrices $W_n^Q, W_n^K, W_n^V \in \mathbb{R}^{d \times d/N}$, and $\mathbf{W}^D \in \mathbb{R}^{d \times d}$ is the output transformation matrix. The multiple attention heads allows our multi-scale Transformer architecture to encode multi-dimensional dependencies among items in S_i.

Non-linearity Injection with Feed-forward Module. In our multi-scale Transformer, we use the point-wise feed-forward network to inject non-linearities into the new generated representations. The non-linear transformation layer is formally represented:

$$\text{PFFN}(\widetilde{\mathbf{H}}^{(l)}) = [\text{FFN}(\tilde{\mathbf{h}}_1^{(l)})^\mathsf{T}, \cdots, \text{FFN}(\tilde{\mathbf{h}}_t^{(l)})^\mathsf{T}] \tag{7}$$
$$\text{FFN}(\mathbf{x}) = \text{GELU}(\mathbf{x}\mathbf{W}_1^{(l)} + \mathbf{b}_1^{(l)})\mathbf{W}_2^{(l)} + \mathbf{b}_2^{(l)},$$

In our feed-forward module, we adopt two layers of non-linear transformation with the integration of intermediate non-linear activation $\text{GELU}(\cdot)$. In addition, $\mathbf{W}_1^{(l)} \in \mathbb{R}^{d \times d_h}, \mathbf{W}_2^{(l)} \in \mathbb{R}^{d_h \times d}, \mathbf{b}_1^{(l)} \in \mathbb{R}^d, \mathbf{b}_2^{(l)} \in \mathbb{R}^d$ are learnable parameters of projection matrices and bias terms. Here, l denotes the l-th multi-scale Transformer layer.

3.2 Customized Hypergraph Learning of Global Multi-Behavior Dependencies

In our MBHT framework, we aim to incorporate long-range multi-behavior dependency into the learning paradigm of evolving user interests. However, it is non-trivial to effectively capture the personalized long-range multi-behavior dependencies. To achieve our goal, we propose to tackle two key challenges in our learning paradigm:

- i) **Multi-Order Behavior-wise Dependency**. The item-wise multi-behavior dependencies are no longer dyadic with the consideration of comprehensive relationships among different types of user behaviors. For example, when deciding to recommend a specific item to users for their potential purchase preference, it would be useful to explore past multi-behavior interactions (*e.g.,* page view, add-to-favorite) between users and this item. Customers are more likely to add their interested products into their favorite item list before making final purchases.

- ii) **Personalized Multi-Behavior Interaction Patterns**. Multi-behavior patterns may vary by users with different correlations across multi-typed user-item interactions. In real-life e-commerce systems, some users like to add many items to their favorite list or cart, if they are interested in, but only a few of them will be purchased later. In contrast, another group of users only tag their interested products as favorite only if they show strong willingness to buy them. Hence, such complex and personalized multi-behavior patterns require our model to preserve the diverse cross-type behaviour dependencies.

To address the above challenges, we build our global multi-behavior dependency encoder upon the hypergraph neural architecture. Inspired by the flexibility of hypergraphs in connecting multiple nodes through a single edge [2, 26], we leverage the hyperedge structure to capture the tetradic or higher-order multi-behavior dependencies over time. Additionally, given the behavior-aware interaction sequence of different users, we construct different hypergraph structures over the sequence S_i ($u_i \in \mathcal{U}$), with the aim of encoding the multi-behavior dependency in a customized manner.

3.2.1 **Item-wise Hypergraph Construction.** In our hypergraph framework, we generate two types of item-wise hyperedge connections corresponding to i) long-range semantic correlations among items; ii) item-specific multi-behavior dependencies across time.

Item Semantic Dependency Encoding with Metric Learning. To encode the time-evolving item semantics and the underlying long-range item dependencies based on the same user interest (*e.g.,* food, outdoor activities), we introduce an item semantic encoder based on a metric learning framework. Specifically, we design the learnable metric $\hat{\beta}_{j,j'}^n$ between items with a multi-channel weight function $\tau(\cdot)$ presented as follows:

$$\beta_{j,j'} = \frac{1}{N} \sum_{n=1}^{N} \hat{\beta}_{j,j'}^n; \mathbf{v}_j = \mathbf{e}_j \oplus \mathbf{b}_j \tag{8}$$

$$\hat{\beta}_{j,j'}^n = \tau(\boldsymbol{w}^n \odot \mathbf{v}_j, \boldsymbol{w}^n \odot \mathbf{v}_{j'}) \tag{9}$$

where $\hat{\beta}_{j,j'}^n$ represents the learnable channel-specific dependency weight between item v_j and $v_{j'}$. We define the weight function $\tau(\cdot)$ as the cosine similarity estimation based on the trainable $\boldsymbol{w}^n$ of n-th representation channel. $(\boldsymbol{w}^n \odot \mathbf{v}_j)$ represents the embedding projection operation. In our item-wise semantic dependency, we perform metric learning under N representation channels (indexed by n). The mean pooling operation is applied to all learned channel-specific item semantic dependency scores (*e.g.,* $\hat{\beta}_{j,j'}^n$), to obtain the final relevance $\beta_{j,j'}$ between item v_j and $v_{j'}$.

Item-wise Semantic Dependency Hypergraph. With the encoded semantic dependencies among different items, we generate the item-wise semantic hypergraph by simultaneously connecting multiple highly dependent items with hyperedges. In particular, we construct a set of hyperedges $\mathcal{E}^p$, where $|\mathcal{E}^p|$ corresponds to the number of unique items in sequence S_i. In the hypergraph $\mathcal{G}_p$ of item-wise semantic dependencies, each unique item will be assigned with a hyperedge in $\epsilon \in \mathcal{E}^p$ to connect top-k semantic dependent items according to the learned item-item semantic dependency score $\beta_{j,j'}$ (encoded from the metric learning component). A_j represents the set of top-k semantic correlated items of a specific item v_j. We define the connection matrix between items and hyperedges as $\mathcal{M}^p \in \mathbb{R}^{J \times |\mathcal{E}^p|}$ in which each entry $m^p(v_j, \epsilon_{j'})$ is:

$$m^p(v_j, \epsilon_{j'}) = \begin{cases} \beta_{j,j'} & v_{j'} \in A_j; \\ 0 & otherwise; \end{cases} \tag{10}$$

where $\epsilon_{j'}$ denotes the hyperedge which is assigned to item $v_{j'}$.

Item-wise Multi-Behavior Dependency Hypergraph. To capture the personalized item-wise multi-behavior dependency in a time-aware environment, we further generate a hypergraph structure $\mathcal{G}_q$ based on the observed multi-typed interactions (*e.g.,* page view, add-to-cart) between user u_i and a specific item v_j at different timestamps. Here, we define $\mathcal{E}^q$ to represent the set of items which have multi-typed interactions with user u_i. In hypergraph $\mathcal{G}_q$, the number of hyperedges is equal to $|\mathcal{E}^q|$. Given that users have diverse multi-behaviour patterns with their interacted items, the constructed multi-behavior dependency hypergraphs vary by

users. To be specific, we generate item-hyperedge connection matrix $\mathcal{M}^q \in \mathbb{R}^{J \times |\mathcal{E}^q|}$ ($m^q \in \mathcal{M}^q$) for hypergraph $\mathcal{G}_q$ as follow:

$$m^q(v_j^b, \epsilon_j^{b'}) = \begin{cases} 1 & v_j^b \in \mathcal{E}_j^q; \\ 0 & otherwise; \end{cases} \tag{11}$$

v_j^b represents that item v_j is interacted with user u_i under the b-th behavior type. $\mathcal{E}_j^q$ denotes the set of multi-typed u_i-v_j interactions.

We further integrate our constructed hypergraph structures $\mathcal{G}_p$ and $\mathcal{G}_q$ by concatenating connection matrices $\mathcal{M}^p$ and $\mathcal{M}^q$ along with the column side. As such, the integrated hypergraph $\mathcal{G}$ is constructed with the concatenated connection matrix $\mathcal{M} \in \mathbb{R}^{J \times (|\mathcal{E}^p| + |\mathcal{E}^q|)}$, *i.e.,* $\mathcal{M} = \mathcal{M}^p \parallel \mathcal{M}^q$. Different behavior-aware interaction sequences result in different hypergraph structures for different users, which allow our MBHT model to encode the personalized multi-behavior dependent patterns in a customized way.

3.2.2 **Hypergraph Convolution Module.** In this module, we introduce our hypergraph message passing paradigm with the convolutional layer, to capture the global multi-behavior dependencies over time. The hypergraph convolutional layer generally involves two-stage information passing [2], *i.e.,* node-hyperedge and hyperedge-node embedding propagation along with the hypergraph connection matrix $\mathcal{M}$ for refining item representations. Particularly, we design our hypergraph convolutional layer as:

$$\mathbf{X}^{(l+1)} = \mathbf{D}_v^{-1} \cdot \mathcal{M} \cdot \mathbf{D}_e^{-1} \cdot \mathcal{M}^\mathsf{T} \cdot \mathbf{X}^{(l)} \tag{12}$$

where $\mathbf{X}^{(l)}$ represents the item embeddings encoded from the l-th layer of hypergraph convolution. Furthermore, $\mathbf{D}_v$ and $\mathbf{D}_e$ are diagonal matrices for normalization based on vertex and edge degrees, respectively. Note that the two-stage message passing by $\mathcal{M} \cdot \mathcal{M}^\mathsf{T}$ takes $O((|\mathcal{E}^p| + |\mathcal{E}^q|) \times J^2)$ calculations, which is quite time-consuming. Inspired by the design in [33], we calculate a matrix $\mathcal{M}'$ by leveraging pre-calculated $\beta_{j,j'}$ to obtain a close approximation representation of $\mathcal{M} \cdot \mathcal{M}^\mathsf{T}$ and thus boost the inference. The detailed process can be found in the supplementary material. We also remove the non-linear projection following [6] to simplify the message passing process. Each item embedding $\mathbf{x}^{(0)}$ in $\mathbf{X}^{(0)}$ is initialized with the behavior-aware self-gating operation as: $\mathbf{x}^{(0)} = (\mathbf{v}_j \oplus \mathbf{b}_j) \odot \text{sigmoid}((\mathbf{v}_j \oplus \mathbf{b}_j) \cdot \mathbf{w} + \mathbf{r})$.

3.3 Cross-View Aggregation

In the forecasting layer of MBHT framework, we propose to fuse the learned item representations from different views: 1) multi-scale behavior-aware sequential patterns with Transformer architecture; 2) personalized global multi-behavior dependencies with Hypergraph framework. To enable this cross-view aggregation in an adaptive way, we develop an attention layer to learn explicit importance for view-specific item embeddings. Formally, the aggregation procedure is presented as follows:

$$\alpha_i = \text{Attn}(\boldsymbol{e}_i) = \frac{\exp(\boldsymbol{a}^\mathsf{T} \cdot \boldsymbol{W}_a \boldsymbol{e}_i)}{\sum_i \exp(\boldsymbol{a}^\mathsf{T} \cdot \boldsymbol{W}_a \boldsymbol{e}_i)} \tag{13}$$

$$\boldsymbol{e}_i \in \{\tilde{\mathbf{h}}_i, \tilde{\mathbf{x}}_i\}; \ \mathbf{g}_i = \alpha_1 \cdot \tilde{\mathbf{h}}_i \oplus \alpha_2 \cdot \tilde{\mathbf{x}}_i$$

where $\tilde{\mathbf{h}}_i$ and $\tilde{\mathbf{x}}_i$ are embeddings from the two views separately for item at the i-th position, and $\boldsymbol{a} \in \mathbb{R}^d$, $\boldsymbol{W}_a \in \mathbb{R}^{d \times d}$ are trainable

Table 1: Statistical information of experimented datasets.

Stats.	Taobao	Retailrocket	IJCAI
# Users	147, 892	11, 649	200, 000
# Items	99, 038	36, 223	808, 354
# Interactions	7, 092, 362	87, 822	13, 072, 940
# Average Length	48.23	14.55	78.58
# Density	5×10^{-6}	1×10^{-6}	7×10^{-7}
# Behavior Types	[buy, cart, fav, pv]	[buy, cart, pv]	[buy, cart, fav, pv]

parameters. Here, to eliminate the over-smooth effect of graph convolution, $\tilde{\mathbf{x}}_i$ is the average of $\mathbf{x}^{(l)}$ across all convolutional layers. Finally, the probability of i-th item in the sequence being item v_j is estimated as: $\hat{y}_{i,j} = \mathbf{g}_i^{\mathsf{T}} \mathbf{v}_j$, where $\mathbf{v}_j$ represents item v_j's embedding.

3.4 Model Learning And Analysis

To fit our multi-behavior sequential recommendation scenario, we utilize the Cloze task [12, 19] as our training objective to model the bidirectional information of item sequence. We describe our Cloze task settings as follows: Considering the multi-behavior sequential recommender systems, we mask all items in the sequence with the target behavior type (*e.g.,* purchase). To avoid the label leak issue, we replace the masked items as well as the corresponding behavior type embeddings with the special token [mask], and leave out masked items in the hypergraph construction in Section 3.2.1. Instead, for masked items, we generate its hypergraph embedding $\mathbf{x}_i$ in Equation 13 by adopting sliding-window average pooling function over hypergraph embeddings of the contextual neighbors surrounding the mask position $(m - q_1, m + q_2)$. The details are presented in Algorithm 1 in the supplementary material. Hence, our model makes prediction on masked items based on the encoded surrounding context embeddings in the behavior interaction sequence. Given the probability estimation function $\hat{y}_{i,j} = \mathbf{g}_i^{\mathsf{T}} \mathbf{v}_j$, we define our optimized objective loss with Cross-Entropy as below:

$$\mathcal{L} = \frac{1}{|T|} \sum_{t \in T, m \in M} -\log\left(\frac{\exp \hat{y}_{m,t}}{\sum_{j \in V} \exp \hat{y}_{m,j}}\right) \quad (14)$$

where T is the set of ground-truth ids for masked items in each batch, M is the set of masked positions corresponding to T, and V is the item set. The time complexity is analyzed in supplementary material.

4 EXPERIMENTS

This section aims to answer the following research questions:

- **RQ1**: How does our MBHT perform as compared to various state-of-the-art recommendation methods with different settings?
- **RQ2**: How effective are the key modules (*e.g.,* multi-scale attention encoder, multi-behavior hypergraph learning) in MBHT?
- **RQ3**: How does MBHT perform to alleviate the data scarcity issue of item sequences when competing with baselines?
- **RQ4**: How do different hyperparameters affect the model performance? (Evaluation results are presented in Appendix A.4).

4.1 Experimental Settings

4.1.1 **Datasets.** We utilize three recommendation datasets collected from real-world scenarios. i) **Taobao**. This dataset is collected from Taobao which is one of the largest e-commerce platforms in China. Four types of user-item interactions are included in this

dataset, *i.e.,* target behaviors-*purchase*; auxiliary behaviors-*add-to-favorites, add-to-cart, page view*. ii) **Retailrocket**. This dataset is generated from an online shopping site-Retailrocket over 4 months, to record three types of user behaviors, *i.e.,* target behaviors-*purchase*; auxiliary behaviors-*page view & add-to-cart*. iii) **IJCAI**. This dataset is released by IJCAI Contest 2015 for the task of repeat buyers prediction. It shares the same types of interaction behaviors with the Taobao dataset. The detailed statistical information of these experimented datasets are summarized in Table 1. Note that different datasets vary by average sequence length and user-item interaction density, which provides diverse evaluation settings.

4.1.2 **Evaluation Protocols.** In our experiments, closely following the settings in [19, 20], we adopt the *leave-one-out* strategy for performance evaluation. For each user, we regard the temporally-ordered last purchase as the test samples, and the previous ones as validation samples. Additionally, we pair each positive sample with 100 negative instances based on item popularity [19]. We utilize three evaluation metrics: *Hit Ratio (HR@N), Normalized Discounted Cumulative Gain (NDCG@N)* and *Mean Reciprocal Rank (MRR)* [21, 25, 30]. Note that larger HR, NDCG and MRR scores indicate better recommendation performance.

4.1.3 **Baselines.** We compare our MBHT with a variety of recommendation baselines to validate the performance superiority.
General Sequential Recommendation Methods.

- **GRU4Rec** [7]. It utilizes the gated recurrent unit as sequence encoder to learn dynamic preference with a ranking-based loss.
- **SASRec** [13]. The self-attention mechanism is leveraged in this method to encode the item-wise sequential correlations.
- **Caser** [20]. This method integrates the convolutional neural layers from both vertical and horizontal views to encode time-evolving user preference of item sequence.
- **HPMN** [17]. It employs a hierarchically structured periodic memory network to model multi-scale transitional information of user sequential behaviors. The incremental updating mechanism is introduced to retain behaviour patterns over time.
- **BERT4Rec** [19]. It uses a bidirectional encoder for modeling sequential information with Transformer. The model is optimized with the Cloze objective, and has produced state-of-the-art performance among sequence learning-based baselines.

Graph-based Sequential Recommender Systems.

- **SR-GNN** [25]. It generates graph structures based on item-item transitional relations in sequences, and conducts graph-based message passing to capture local and global user interests.
- **GCSAN** [28]. It empowers the self-attention mechanism with with a front-mounted GNN structure. The attentive aggregation is performed over the encoded graph embeddings.
- **HyperRec** [21]. It designs sequential hypergraphs to capture evolving users' interests and regards users as hyperedges to connect interacted items, so as to model dynamic user preferences.
- **SURGE** [1]. It adopts metric learning to build personalized graphs and uses hierarchical attention to capture multi-dimensional user interests in the graph.

Multi-Behavior Recommendation Models.

- **BERT4Rec-MB** [19]. We enhance the BERT4Rec method to handle the dynamic multi-behavior context by injecting behavior type representations into the input embeddings for self-attention.

Table 2: The performance of our method and the best performed baseline are presented with bold and underlined, respectively. Superscript ∗ indicates the significant improvement between our MBHT and the best performed baseline with p value < 0.01.

Model	Taobao					Retailrocket					IJCAI				
	HR@5	NDCG@5	HR@10	NDCG@10	MRR	HR@5	NDCG@5	HR@10	NDCG@10	MRR	HR@5	NDCG@5	HR@10	NDCG@10	MRR
General Sequential Recommendation Methods															
Caser	0.082	0.058	0.123	0.071	0.070	0.632	0.539	0.754	0.578	0.535	0.134	0.092	0.167	0.104	0.109
HPMN	0.162	0.130	0.219	0.141	0.139	0.664	0.633	0.711	0.587	0.602	0.144	0.085	0.197	0.124	0.123
GRU4Rec	0.147	0.105	0.209	0.125	0.118	0.640	0.575	0.708	0.597	0.572	0.141	0.100	0.200	0.119	0.113
SASRec	0.150	0.110	0.206	0.128	0.123	0.669	0.644	0.689	0.650	0.645	0.146	0.110	0.191	0.124	0.122
BERT4Rec	0.198	0.153	0.254	0.171	0.163	0.808	0.670	0.881	0.694	0.639	0.297	0.220	0.402	0.253	0.227
Graph-based Sequential Recommender Systems															
SR-GNN	0.102	0.071	0.153	0.087	0.086	0.848	0.780	0.891	0.793	0.767	0.072	0.048	0.118	0.062	0.064
GCSAN	0.217	0.160	0.305	0.188	0.173	0.872	0.846	0.890	0.851	0.842	0.119	0.086	0.175	0.104	0.101
HyperRec	0.145	0.130	0.224	0.133	0.129	0.860	0.705	0.833	0.820	0.816	0.140	0.109	0.236	0.144	0.132
SURGE	0.122	0.078	0.193	0.100	0.093	0.906	0.879	0.878	0.887	0.870	0.226	0.159	0.322	0.190	0.171
Multi-Behavior Recommendation Models															
BERT4Rec-MB	0.211	0.169	0.263	0.186	0.178	0.875	0.858	0.889	0.863	0.857	0.257	0.189	0.342	0.216	0.197
MB-GCN	0.185	0.103	0.309	0.143	0.149	0.878	0.735	0.844	0.752	0.739	0.218	0.145	0.335	0.182	0.177
NMTR	0.125	0.082	0.174	0.097	0.103	0.858	0.697	0.827	0.724	0.741	0.109	0.076	0.184	0.099	0.106
MB-GMN	0.196	0.115	0.319	0.154	0.151	0.901	0.762	0.853	0.830	0.822	0.235	0.161	0.337	0.193	0.176
MBHT	**0.323***	**0.257***	**0.405***	**0.283***	**0.262***	**0.956***	**0.950***	**0.931***	**0.933***	**0.929***	**0.346***	**0.268***	**0.437***	**0.297***	**0.272***
# Improve	48.84%	52.07%	26.95%	50.53%	47.19%	5.17%	8.07%	6.36%	5.19%	6.78%	16.50%	21.82%	8.71%	17.39%	19.82%

- **MB-GCN** [11]. This model is built upon the graph convolutional layer to refine user/item embeddings through the behavior-aware message passing on the user-item interaction graph.
- **NMTR** [3]. It defines the behavior-wise cascading relationships to model the dependency among different types of behaviors under a multi-task learning paradigm.
- **MB-GMN** [27]. It employs a graph meta network to capture personalized multi-behavior signals and model the diverse multi-behavior dependencies. It generates state-of-the-art performance among different multi-behavior recommendation methods.

4.1.4 **Hyperparameter Settings.** In MBHT model, we search the number of hypergraph propagation layers from {1,2,3,4}. The number of multi-head channels is set as 2 for both self-attention and metric learning components. The value of k for constructing item-wise semantic dependency hypergraph is tuned from [4, 6, 8, 10, 12, 14]. Considering that datasets vary by average sequence length, the multi-scale setting parameters (C, p_1, p_2) are searched amongst the value range of ([20,4,20],[20,8,40],[40,4,20],[40,8,40]).

4.2 Performance Evaluation (RQ1)

We report the detailed performance comparison on different datasets in Table 2 and summarize the observations as followed:

- The proposed MBHT consistently outperforms all types of baselines by a significant margin on different datasets. The performance improvements can be attributed from: i) Through the multi-scale behavior-aware Transformer, MBHT is able to capture the behavior-aware item transitional patterns from fine-grained to coarse-grained time granularities. ii) With the hypergraph neural network for multi-behavior dependency learning, we endow MBHT with the capability of capturing long-range item correlations across behavior types over time.
- By jointly analyzing the results across different datasets, we can observe that our MBHT is robust to different recommendation scenarios with various data characteristics, such as average sequence length and user-item interaction density, reflecting various user behaviour patterns in many online platforms.
- Graph-based sequential recommendation methods (*e.g.,* SR-GNN, GCSAN, SURGE) perform worse than general sequential baselines (*e.g.,* BERT4Rec, HPMN) on IJCAI dataset with longer item

Table 3: Ablation study with key modules.

| Model Variants	Taobao		Retailrocket		IJCAI	
	HR@5	NDCG@5	HR@5	NDCG@5	HR@5	NDCG@5
MBHT	**0.323**	**0.257**	**0.956**	**0.950**	**0.346**	**0.268**
(-) MB-Hyper	0.261	0.206	0.883	0.861	0.320	0.249
(-) ML-Hyper	0.271	0.212	0.898	0.874	0.328	0.256
(-) Hypergraph	0.246	0.194	0.813	0.839	0.301	0.234
(-) MS-Attention	0.253	0.200	0.816	0.832	0.329	0.256

sequences. The possible reason is that passing message between items over the generated graph structures based on their directly transitional relations can hardly capture the long-term item dependencies. However, the performance superiority of GNN-based models can be observed on Retailrocket with shorter item sequences. In such cases, modeling of short-term item transitional regularities is sufficient for capturing item dependencies.

- BERT4Rec-MB outperforms BERT4Rec in most evaluation cases. In addition, it can be found that multi-behavior recommendation models (*e.g.,* MB-GCN, MB-GMN) achieve comparable performance to other baselines. These observations indicate the effectiveness of incorporating multi-behavior context into the learning process of user preference. With the effective modeling of dynamic multi-behaviour patterns from both short-term and long-term perspectives, our MBHT is more effective than those stationary multi-behavior recommendation approaches.

4.3 Ablation Study (RQ2)

4.3.1 **Effects of Key Components.** We firstly investigate the effectiveness of different components of our MBHT from both Transformer and Hypergraph learning views. Specifically, we generate four variants and make comparison with our MBHT method:

- *(-) MB-Hyper.* This variant does not include the hypergraph of item-wise multi-behavior dependency to capture the long-range cross-type behavior correlations.
- *(-) ML-Hyper.* In this variant, we remove the hypergraph message passing over the hyperedges of item semantic dependence (encoded with the metric learning component).
- *(-) Hypergraph.* This variant disables the entire hypergraph item-wise dependency learning, and only relies on the multi-scale Transformer to model the sequential behavior patterns.
- *(-) MS-Attention.* For this variant, we replace our multi-scale attention layer with the original multi-head attentional operation.

(a) Taobao   (b) IJCAI   (c) Retailrocket
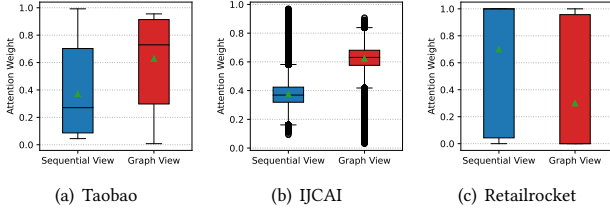
Figure 3: Distributions of the learned attentive view-specific contributions. Green triangles and black line in the showed boxes denote the mean and median values, respectively.


(a) Taobao   (b) IJCAI
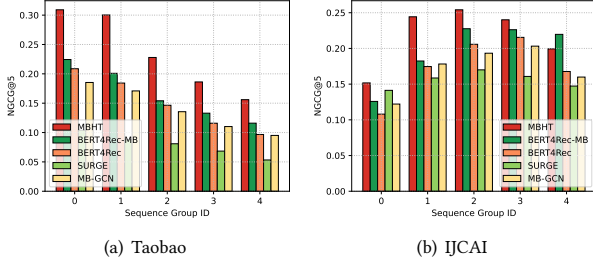
Figure 4: Performance *w.r.t* different sequence lengths.

From the reported results in Table 3, we summarize the following observations to show the rationality of our model design. 1) With the incorporation of hypergraph-based dependency learning on either item-wise latent semantic ((-) MB-Hyper) or dynamic multi-behavior correlations ((-) ML-Hyper), MBHT can further boost the recommendation performance. 2) Comparing with the vanilla multi-head attention ((-) MS-Attention), the effectiveness of our multi-scale low-rank self-attention can be validated.

4.3.2 **Contribution of Learning Views.** We further investigate the contribution of sequential and hypergraph learning views, by presenting the distributions of our learned importance scores of $\mathbf{h}_i$ and $\mathbf{x}_i$ in Figure 3. In particular, $\mathbf{h}_i$ preserves multi-scale behavior dynamics of diverse user preference and $\mathbf{x}_i$ encodes the global multi-behavior dependencies. We can observe that hypergraph learning view contributes more to the effective modeling of dynamic multi-behavior patterns with longer item sequences (*e.g.,* Taobao and IJCAI dataset). This further confirms the efficacy of our behavior-aware hypergraph learning component in capturing the long-range item dependencies in multi-relational sequential recommendation.

4.4 Model Benefit Study (RQ3)

4.4.1 **Performance *w.r.t* Sequence Length.** To further study the robustness of our model, we evaluate MBHT on item sequences with different length. Specifically, we split users into five groups in terms of their item sequences and conduct the performance comparison on each group of users. From results presented in Figure 4, MBHT outperforms several representative baselines not only on the shorter item interaction sequences, but also on the longer item sequences. It indicates that our recommendation framework is enhanced by injecting the behavior-aware short-term and long-term dependencies (from locally to globally) into the sequence embeddings. Such data scarcity issue is hard to be alleviated purely from the general and GNN-based sequential recommender systems.

(a) Taobao   (b) IJCAI
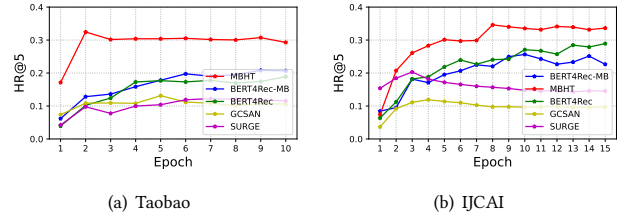
Figure 5: Training curves evaluated by testing Hit Rate.
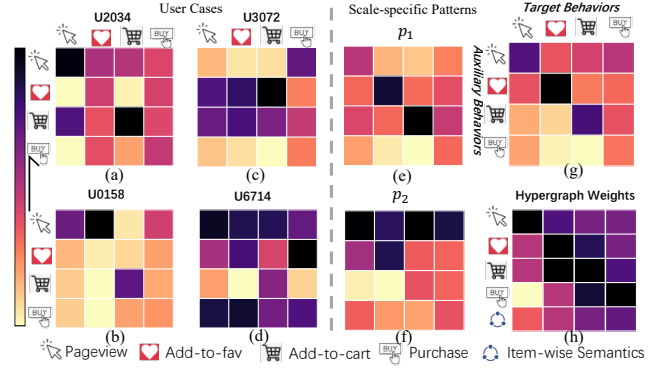
Figure 6: Case studies with cross-type behavior dependencies.

4.4.2 **Model Convergence Study.** We further investigate the convergence property of our MBHT and various sequential and graph-based temporal recommendation methods in Figure 5. Along the model training process, MBHT achieves faster convergence rate compared with most competitive methods. For example, MBHT obtains its best performance at epoch 2 and 8, while BERT4Rec and BERT4Rec-MB take 10 and 15 epochs to converge on Taobao and IJCAI datasets, respectively. This observation suggests that exploring the augmented multi-behavior information from both sequential and hypergraph views can provide better gradient to guide the model optimization in sequential recommendation.

4.5 Case Study

In this section, we conduct further model analysis with case study to show the model interpretation for multi-behavior dependency modeling. In particular, we show user-specific cross-type behavior dependencies in Figure 6 (a)-(d). Each 4×4 dependency matrix is learned from our multi-scale Transformer. We compute the item-item correlations by considering their behavior-aware interactions in the specific sequence S_i of user u_i. Figure 6 (e)-(f) show the scale-specific multi-behavior dependencies with the scales of p_1 and p_2 in our multi-scale modeling of behavior-aware sequential patterns. In Figure 6 (g), we show the overall relevance scores among different types of behaviors in making final forecasting on target behaviors. Additionally, our hypergraph-based item dependencies ($\boldsymbol{M} \cdot \boldsymbol{M}^{\mathsf{T}}$) are shown in Figure 6 (h), such as hypergraph-based i) behavior-aware item relevance; and ii) item-wise semantic dependence.

5 RELATED WORK

Sequential Recommendation. Earlier studies solve the next-item recommendation using the Markov Chain-based approaches to model item-item transitions [5, 18]. In recent years, many efforts

have been devoted to proposing neural network-enhanced sequential recommender systems to encode the complex dependencies among items from different perspectives. For example, recurrent neural network in GRU4Rec [7] and convolutional operations in Caser [20]. Inspired by the strength of Transformer, SASRec [13] and BERT4Rec [19] are built upon the self-attention mechanism for item-item relation modeling. Furthermore, recently emerged graph neural networks produce state-of-the-art performance by performing graph-based message passing among neighboring items, in order to capture sequential signals, *e.g.*, SR-GNN [25], MTD [8], MA-GNN [16] and SURGE [1]. However, most of those methods are specifically designed for singular type of interaction behaviors, and cannot handle diverse user-item relationships.

Hypergraph Learning for Recommendation. Motivated by expressiveness of hypergraphs [2, 32], hypergraph neural networks are utilized in several recent recommender systems to model high-order relationships, such as multi-order item correlations in HyperRec [21], global user dependencies in HCCF [26], high-order social relationships in MHCN [33], and item dependencies with multi-modal features in HyperCTR [4]. Following this research line, this work integrates the hypergraph neural architecture with Transformer architecture to encode behavior-aware sequential patterns from local to global-level for comprehensive behavior modeling.

Multi-Behavior Recommender Systems. There exist recently developed multi-behavior recommender systems for modeling user-item relation heterogeneity [3, 11, 23, 29, 35]. For example, NMTR [3] is a multi-task recommendation framework with the predefined behavior-wise cascading relationships. Motivated by the strength of GNNs, MBGCN [11], MBGMN [27], MGNN [35] are developed based on the graph-structured message passing over the generated multi-relational user-item interaction graphs. Nevertheless, none of those approaches considers the time-evolving multi-behaviour user preference. To fill this gap, our MBHT model is able to capture both short-term and long-term multi-behavior dependencies with a hypergraph-enhanced transformer architecture.

6 CONCLUSION

In this paper, we present a new sequential recommendation framework MBHT which explicitly captures both short-term and long-term multi-behavior dependencies. MBHT designs a multi-scale Transformer to encode the behavior-aware sequential patterns at both fine-grained and coarse-grained levels. To capture the global cross-type behavior dependencies, we empower MBHT with a multi-behavior hypergraph learning component. Empirical results on several real-world datasets validate the strengths of our MBHT when competing with state-of-the-art recommendation methods.

ACKNOWLEDGMENTS

REFERENCES

[1] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, et al. 2021. Sequential Recommendation with Graph Neural Networks. In *SIGIR*. 378–387.
[2] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *AAAI*, Vol. 33. 3558–3565.
[3] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, et al. 2019. Neural multi-task recommendation from multi-behavior data. In *ICDE*. IEEE, 1554–1557.
[4] Li He, Hongxu Chen, Dingxian Wang, Shoaib Jameel, Philip Yu, et al. 2021. Click-Through Rate Prediction with Multi-Modal Hypergraphs. In *CIKM*. 690–699.
[5] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *ICDM*. IEEE, 191–200.
[6] Xiangnan He, Kuan Deng, Xiang Wang, et al. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.
[7] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *ICLR*.
[8] Chao Huang, Jiahui Chen, Lianghao Xia, Yong Xu, Peng Dai, Yanqing Chen, Liefeng Bo, Jiashu Zhao, and Jimmy Xiangji Huang. 2021. Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation. In *AAAI*.
[9] Chao Huang, Xian Wu, Xuchao Zhang, Chuxu Zhang, Jiashu Zhao, Dawei Yin, and Nitesh V Chawla. 2019. Online purchase prediction via multi-scale modeling of behavior dynamics. In *KDD*. 2613–2622.
[10] Hao Jiang et al. 2020. What aspect do you like: Multi-scale time-aware user interest modeling for micro-video recommendation. In *MM*. 3487–3495.
[11] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, et al. 2020. Multi-behavior recommendation with graph convolutional networks. In *SIGIR*. 659–668.
[12] Taegwan Kang, Hwanhee Lee, Byeongjin Choe, and Kyomin Jung. 2021. Entangled bidirectional encoder to autoregressive decoder for sequential recommendation. In *SIGIR*. 1657–1661.
[13] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. IEEE, 197–206.
[14] Nikita Kitaev et al. 2020. Reformer: The efficient transformer. In *ICLR*.
[15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, et al. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*. 10012–10022.
[16] Chen Ma, Liheng Ma, Yingxue Zhang, et al. 2020. Memory augmented graph neural networks for sequential recommendation. In *AAAI*, Vol. 34. 5045–5052.
[17] Kan Ren, Jiarui Qin, et al. 2019. Lifelong sequential modeling with personalized memorization for user response prediction. In *SIGIR*. 565–574.
[18] Steffen Rendle, Christoph Freudenthaler, et al. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*. 811–820.
[19] Fei Sun, Jun Liu, et al. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*. 1441–1450.
[20] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.
[21] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *SIGIR*. 1101–1110.
[22] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-Attention with Linear Complexity. arXiv:2006.04768 [cs.LG]
[23] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive Meta Learning with Behavior Multiplicity for Recommendation. In *WSDM*. 1120–1128.
[24] Liang Wu, Diane Hu, Liangjie Hong, et al. 2018. Turning clicks into purchases: Revenue optimization for product search in e-commerce. In *SIGIR*. 365–374.
[25] Shu Wu, Yuyuan Tang, Yanqiao Zhu, et al. 2019. Session-based recommendation with graph neural networks. In *AAAI*, Vol. 33. 346–353.
[26] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Xiangji Huang. 2022. Hypergraph Contrastive Collaborative Filtering. *arXiv preprint arXiv:2204.12200*.
[27] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. 2021. Graph meta network for multi-behavior recommendation. In *SIGIR*. 757–766.
[28] Chengfeng Xu, Pengpeng Zhao, et al. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation. In *IJCAI*, Vol. 19. 3940–3946.
[29] Haoran Yang, Hongxu Chen, Lin Li, S Yu Philip, and Guandong Xu. 2021. Hyper Meta-Path Contrastive Learning for Multi-Behavior Recommendation. In *ICDM*. IEEE, 787–796.
[30] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge Graph Contrastive Learning for Recommendation. *arXiv preprint arXiv:2205.00976* (2022).
[31] Shaowei Yao and Xiaojun Wan. 2020. Multimodal transformer for multimodal machine translation. In *ACL*. 4346–4350.
[32] Jaehyuk Yi and Jinkyoo Park. 2020. Hypergraph convolutional recurrent neural network. In *KDD*. 3366–3376.
[33] Junliang Yu, Hongzhi Yin, et al. 2021. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *WWW*. 413–424.
[34] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.
[35] Weifeng Zhang, Jingwen Mao, Yi Cao, and Congfu Xu. 2020. Multiplex graph neural networks for multi-behavior recommendation. In *CIKM*. 2313–2316.
[36] Xiangmin Zhou, Dong Qin, Xiaolu Lu, Lei Chen, and Yanchun Zhang. 2019. Online social media recommendation over streams. In *ICDE*. IEEE, 938–949.

A SUPPLEMENTARY MATERIAL

In our supplemental material, we first summarize the learning process of our MBHT framework in Algorithm 1 and conduct the model time complexity analysis. Then, we present our strategy to simplify the implementation of our hypergraph-based embedding propagation, so as to improve the model efficiency.

A.1 The Learning Process of MBHT

Algorithm 1: The forward propagation flow of MBHT

Input : The behavior-aware interaction sequence S_i for user u_i with mask tokens at positions M and with true labels T.
$S_i = [(v_{i,1}, b_{i,1}), ..., [\text{mask}], ..., (v_{i,J}, b_{i,J})]$.

Output : The estimated probability for user u_i interacting with ground-truth items T at the time step positions M.

1 *Multi-Scale Transformer View*;

2 Inject positional and behavior signals into Transformer inputs: $\mathbf{H} \leftarrow [\mathbf{h}_1, ..., \mathbf{h}_{J+1}]$, $\mathbf{h}_j \leftarrow \mathbf{e}_j \oplus \mathbf{p}_j \oplus \mathbf{b}_j$;

3 Perform multi-scale attention under multi-scale settings C, p_1 and p_2 according to Equation 2-4:
$\widetilde{\mathbf{H}} \leftarrow f(\widehat{\mathbf{H}} \parallel \mathbf{H}^{p_1} \parallel \mathbf{H}^{p_2})$;

4 Perform point-wise feed-forward to inject non-linearity:
$\widetilde{\mathbf{H}}^{(l)} \leftarrow [\text{FFN}(\tilde{\mathbf{h}}_1^{(l)})^\mathsf{T}, \cdots, \text{FFN}(\tilde{\mathbf{h}}_t^{(l)})^\mathsf{T}]$;

5 *Hypergraph View*;

6 Model item-wise semantic dependencies with multi-channel metric learning: $\beta_{j,j'} \leftarrow \frac{1}{N} \sum_{n=1}^{N} \hat{\beta}_{j,j'}^n$;

7 Construct customized hyperedges according to Equation 10-11: $\mathcal{M} \leftarrow \mathcal{M}^p \parallel \mathcal{M}^q$;

8 Apply hypergraph convolutional function to aggregate information from the graph:
$\mathbf{X}^{(l+1)} \leftarrow \mathbf{D}_v^{-1} \cdot \mathcal{M} \cdot \mathbf{D}_e^{-1} \cdot \mathcal{M}^\mathsf{T} \cdot \mathbf{X}^{(l)}$;

9 *Fusion and Prediction*;

10 **foreach** $m \in M, t \in T$ **do**

11 Generate hypergraph-view embedding for mask position m using sliding-window contextual pooling:
$\tilde{\mathbf{x}}_m \leftarrow \text{mean}(\tilde{\mathbf{x}}_{m-q_1}, ..., \tilde{\mathbf{x}}_{m+q_2})$;

12 Apply attentive cross-view aggregation to fuse the information according to Equation 13:
$\mathbf{g}_m \leftarrow \alpha_1 \cdot \tilde{\mathbf{h}}_m \oplus \alpha_2 \cdot \tilde{\mathbf{x}}_m$;

13 Calculate the probability of item at m being v_t:
$\hat{y}_{m,t} \leftarrow \mathbf{g}_m^\mathsf{T} \mathbf{v}_t$;

14 **end**

15 **return** $[\hat{y}_{m_1,t_1}, \hat{y}_{m_2,t_2}, ...]$;

A.2 Time Complexity Analysis

This section conducts the time complexity analysis of our MBHT framework as follows. (1) For the multi-scale Transformer view, with our low-rank self-attention layer, we significantly reduce the computational cost from $O(L \times d \times ((\frac{J}{C})^2 + (\frac{J}{p_1})^2 + (\frac{J}{p_2})^2))$ to approximate the linear time complexity $O(3LdJ)$, given that C (low-rank

scale), p_1, p_2 (resolution scales) $\ll J$ [22]. J denotes the length of item sequence. (2) For the hypergraph learning view, the semantic metric learning takes $O(J^2 d)$ complexity, and the hypergraph convolutional function takes $O(LJd^2)$. Based on the above discussion, the overall time complexity of our MBHT is $O(3LJd + LJd^2 + J^2 d)$, which is comparable to state-of-the-art baselines.

A.3 Simplifying Hypergraph Message Passing

With the consideration of high computational cost during the message passing in our hypergraph learning framework, we propose to simplify the propagation scheme with the learnable embedding projection. Formally, we rewrite the two-stage (node-hyperedge-node) message propagation process $\mathcal{M} \cdot \mathcal{M}^\mathsf{T}$ as follows:

$$(\mathcal{M} \cdot \mathcal{M}^\mathsf{T})_{ij} = \sum_{k=1}^{n} \mathcal{M}_{ik} \mathcal{M}_{kj}^\mathsf{T} = \mathcal{M}_{(i)} \mathcal{M}^{\mathsf{T}(j)} = \mathcal{M}_{(i)} \mathcal{M}_{(j)} \quad (15)$$

where $\mathcal{M}_{(i)}$ and $\mathcal{M}_{(j)}$ denotes the i-th row and j-th column vector, respectively. Based on our item-wise semantic dependence and multi-behavior correlations, our hypergraph-guided information propagation can be presented as follows:

$$\mathcal{M}_{(i)} \mathcal{M}_{(j)} = \sum_{k=1}^{|\mathcal{E}^q|} \mathcal{M}_{i,k} \mathcal{M}_{j,k} + \sum_{m=1}^{|\mathcal{E}^p|} \mathcal{M}_{i,m} \mathcal{M}_{j,m} \quad (16)$$

where $\mathcal{M}_{i,k}$ and $\mathcal{M}_{i,m}$ denote the value of hypergraph connection matrix with item-wise semantic dependency and multi-behavior correlations of item v_i, respectively. Here, we first simplify the multiplication operations with the multi-behavior dependency hyperedges. Specifically, the same item ($v_i = v_j$) interacted with different behavior types over time will be connected to the same hyperedge ϵ^q, i.e., $\mathcal{M}_{i,\epsilon^q} = \mathcal{M}_{j,\epsilon^q} = 1$. Since each item can only be connected to one hyperedge, for $\epsilon' \neq \epsilon^q$, $\mathcal{M}_{i,\epsilon'} = \mathcal{M}_{j,\epsilon^q} = 0$. This indicates that if $v_i \neq v_j$, there exists no multi-behavior dependency hyperedge ϵ' such that $\mathcal{M}_{i,\epsilon'} \mathcal{M}_{j,\epsilon'} = 1$, since different items cannot be connected to the same hyperedge. Hence, we can isolate the influence of the multi-behavior dependency hyperedges as follows:

$$\sum_{\epsilon'=1}^{|\mathcal{E}^q|} \mathcal{M}_{i,\epsilon'} \mathcal{M}_{j,\epsilon'} = \begin{cases} 1 & \text{if } v_i = v_j \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

We further investigate the item-wise semantic dependency hyperedges in Equation 16 for simplifying hypergraph convolutional operations. Note that, $\mathcal{M}_{i,m}$ denotes the cosine similarity between item v_i and v_m that belong to the hyperedge m. Generally, the computation $\sum_{m=1}^{|\mathcal{E}^p|} \mathcal{M}_{im} \mathcal{M}_{jm}$ can be divided into three cases depending on the hyperedge m: i) behavior-aware self-connection, if $v_i = v_j$ and m is the hyperedge assigned to this item; ii) first-order similarity, if $v_i \neq v_j$ and m is assigned to v_i or v_j; iii) second-order similarity, if $v_i \neq v_j$ and m is not assigned to either v_i or v_j.

We leverage the pre-calculated behavior-aware semantic similarities $\beta_{i,j}$ between items for the first and second cases. Here $\beta_{i,j}$ is truncated from the top-k value to be consistent with the semantic dependency hyperedges. We denote the values of the third case by $w_{i,j}$. Therefore, based on the above discussions, for the first case, we have: $\sum_{m=1}^{|\mathcal{E}^p|} \mathcal{M}_{im} \mathcal{M}_{jm} = \beta_{i,j} + w_{i,j}$; for the second case, we have $\sum_{m=1}^{|\mathcal{E}^p|} \mathcal{M}_{im} \mathcal{M}_{jm} = \beta_{i,j} + \beta_{j,i} + w_{i,j}$. Since the computation of the second-order complexity is time-consuming, and a number

Table 4: Comparison between two implementation of hypergraph message passing schemes, *i.e.,* the original embedding propagation based on hypergraph connection matrices, and the simplified message passing schema. The recommendation accuracy is measured by Recall@5 and model computational cost is measured by the running time of each epoch.

	Taoabo		Retailrocket		IJCAI	
	Recall@5	Epoch Time	Recall@5	Epoch Time	Recall@5	Epoch Time
Origin	0.326	86.43	0.959	4.58	0.352	133.28
Simplified	0.323	38.05	0.956	2.11	0.346	65.2

(a) Sensitivity to attention scales   (b) Sensitivity to sim-group length
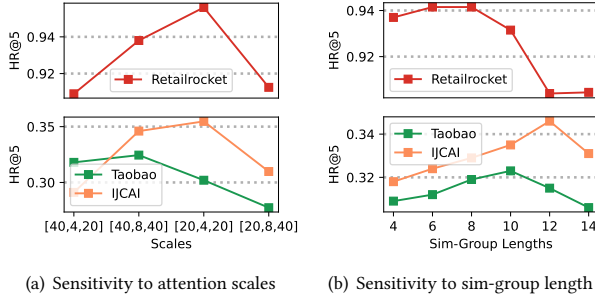
Figure 7: Hyperparameter study of MBHT framework.

of such values are slight due to the top-k truncation, we replace $w_{i,j}$ by a hyperparameter w_0 to obtain a close approximation $\mathcal{M}'$ for $\mathcal{M}\mathcal{M}^\mathsf{T}$. Formally, $\mathcal{M}' \in \mathbb{R}^{J \times J}$ is generated as follows:

$$\mathcal{M}' = C \oplus A \oplus W \tag{18}$$

where $C_{i,j} = 1$ if $v_i = v_j$, otherwise $C_{i,j} = 0$, and $A_{i,j} = \beta_{i,j}$ if $v_i = v_j$, otherwise $A_{i,j} = \beta_{i,j} + \beta_{j,i}$. $W_{i,j} = w_0$. In our experiments, w_0 is searched among $[0.05, 0.1, 0.15, 0.2]$. Finally, we express the light version of hypergraph convolution function below:

$$\mathbf{X}^{(l+1)} = \mathbf{D}^{-1} \cdot \mathcal{M}' \cdot \mathbf{X}^{(l)} \tag{19}$$

We conduct empirical experiments to further evaluate the model performance of our simplified hypergraph convolution mechanism. We report the evaluation results in Table 4. The model training is performed on a single GTX3090 GPU for running time evaluation. We can observe that our simplified hypergraph-based message

passing scheme only lead to slightly performance degradation and improve the model efficiency with lower computational cost. The potential reasons are: i) a large number of second-order similarity are slight, since scores are truncated from top-k and ii) the convolution process inherently takes into account high-order connectivity. At the same time, simplifying the convolutional function brings a lot enhancement on inference speed, since it eliminates the original $O((|\mathcal{E}^p| + |\mathcal{E}^q|) \times J^2)$ calculations and $\mathcal{M}'$ can be easily built by leveraging pre-calculated values and data pre-processing.

A.4 Hyperparameter Study (RQ4)

We conduct experiments to analyze the influence of key hyperparameters in our MBHT framework and report results in Figure 7.

Impact of Multi-Scale Settings. We search multi-scale setting parameters (C, p_1, p_2) among the range $\{[20, 4, 20], [20, 8, 40], [40, 4, 20], [40, 8, 40]\}$. We present the observations as followed:

- The best performance on Retailrocket and Taobao datasets can be achieved with $(p_1, p_2) = (4, 20)$ and $(p_1, p_2) = (8, 40)$ given the difference of average sequence length.

- For the low-rank projection scale parameter C, we can notice that projecting original self-attentive sequence embedding space into $\frac{J}{20}$ channels can bring the best performance on IJCAI and Retailrocket datasets. For Taobao dataset, we can observe that $\frac{J}{40}$ performs better than $\frac{J}{20}$, which indicates that dense user-item interaction data may need less low-rank projection channels for better sequential pattern encoding.

Impact of Item-wise Semantic Dependency Set. Our hypergraph item dependency encoder investigates the latent semantic correlations among different items. We search the top-k semantic dependent items from {4,6,8,10,12,14} for global message passing through the item-wise semantic hyperedges. Observations are:

- The best settings of k is proportionally to the average sequence length of different datasets. It indicates that larger hypergraph propagation scope is better for modeling longer item sequences.

- Increasing the number of connected items through hyperedges may firstly boost the performance at the early stage, and then lead to performance degradation by involving noise during the hypergraph-based embedding propagation.