# Graph Meta Network for Multi-Behavior Recommendation

Lianghao Xia
South China University of Technology
cslianghao.xia@mail.scut.edu.cn

Yong Xu
South China University of Technology
yxu@scut.edu.cn

Chao Huang*
JD Finance America Corporation
chaohuang75@gmail.com

Peng Dai
JD Finance America Corporation
peng.dai@jd.com

Liefeng Bo
JD Finance America Corporation
liefeng.bo@jd.com

## ABSTRACT

Modern recommender systems often embed users and items into low-dimensional latent representations, based on their observed interactions. In practical recommendation scenarios, users often exhibit various intents which drive them to interact with items with multiple behavior types (*e.g.*, click, tag-as-favorite, purchase). However, the diversity of user behaviors is ignored in most of existing approaches, which makes them difficult to capture heterogeneous relational structures across different types of interactive behaviors. Exploring multi-typed behavior patterns is of great importance to recommendation systems, yet is very challenging because of two aspects: i) The complex dependencies across different types of user-item interactions; ii) Diversity of such multi-behavior patterns may vary by users due to their personalized preference. To tackle the above challenges, we propose a Multi-Behavior recommendation framework with Graph Meta Network to incorporate the multi-behavior pattern modeling into a meta-learning paradigm. Our developed MB-GMN empowers the user-item interaction learning with the capability of uncovering type-dependent behavior representations, which automatically distills the behavior heterogeneity and interaction diversity for recommendations. Extensive experiments on three real-world datasets show the effectiveness of MB-GMN by significantly boosting the recommendation performance as compared to various state-of-the-art baselines. The source code is available at *https://github.com/akaxlh/MB-GMN*.

## CCS CONCEPTS

• **Information systems → Recommender systems**.

---

*Chao Huang is the corresponding author.

---

## 1 INTRODUCTION

Recommender systems have played an important role in meeting users' personalized interests and alleviating the information overload for various applications, ranging from online review sites [27], location-based recommendation services [1, 10], and online retailing platforms [12, 19]. To forecast user preference from observed user behavior data, many efforts have been devoted to Collaborative Filtering (CF) techniques (*e.g.*, matrix factorization methods [23]). The key idea of CF paradigm is to consider user historical interactions, and make recommendations based on their potential common preferences with vectorized user/item representations [16].

With the recent advancements in deep learning, many neural network techniques have been proposed to augment collaborative filtering architecture, with the modeling of complex interactive patterns between users and items. For instance, NCF [16] and DMF [37] introduce the Multi-layer perceptron to enable the CF framework with the capability of learning non-linear feature interactions. In addition, by stacking multiple autoencoder networks, high-dimensional sparse user behavior data can be mapped into the low-dimensional dense representations under the reconstruction-based learning prototype [8, 25]. Inspired by the recent success of graph neural networks, several studies are proposed to model the user-item interactions with graph relation encoder (*e.g.*, Graph Convolutional Network [40] or Graph Attention Network [28]).

Nevertheless, a common drawback of the above methods is that only single type of behavior is considered to characterize user-item collaborative relations, which may not comprehensively reflect the multi-dimensional users' preference in many real-world scenarios. In fact, we need to deal with more complex user-item relational structures–multi-behavior interaction scenario in which each interaction between user and item is naturally exhibited with relationship diversity [13, 36]. For example, there may exist multiple relations (*e.g.*, click, add-to-cart, add-to-favorite and purchase) between the same pair of user and item in e-commence platforms [3]. Each individual view of behavior type could provide rich information to characterize users' preference from different dimensions. Some users might like to add their interested items into the favorite item set before placing the order, while others always make purchases once they add items into the cart. This motivates us to explore the multi-behavior interactive patterns in modeling users' preference that yields more accurate recommendations.

While it is desirable to make use of other types of behaviors in assisting the recommendation task, it is non-trivial to encode such multi-behavioral collaborative relations. Towards this end, the

technical challenges mainly come from two perspectives:

**Modeling with Behavior Heterogeneity**. It is challenging to learn a comprehensive representation which can comprehensively reflect preference patterns of user-item interaction from different behavior types. Different behavior type view can present complementary knowledge for learning users' interests and interweave with each other in complex ways. Furthermore, each type of user behavior may exhibit pairwise dependencies or even high-level cross-type behavior correlations. Hence, in order to learn meaningful multi-behavior representations from type-specific, inter-type behavior relations, an effective cross-type interactive feature learning model with behavior pattern fusion is a necessity in solving multi-behavior recommendation problem.

**Learning with Interaction Diversity**. The dependencies across different types of user-item interactions are complex since user's type-specific behaviors are likely to be dependent because of various factors [31]. In practical scenarios, users often exhibit their different correlated behavioral patterns due to their personal specialty and habits, *e.g.*, some people prefer to add items into their favorite lists only if they will buy them with high probabilities. In contrast, some one may like to add many different items into his/her favorite list but only makes sporadic purchases. The interdependencies between multiple types of behaviors vary by users. Although a handful of studies attempt to learn users' interests from their different types of user behaviors [9, 13, 21], they merely consider the correlations between different types of user-item interactions under the same representation space, which can hardly capture personalized characteristics from users' multi-behavior interactions. Therefore, a sufficiently generalized knowledge transfer framework is required, to distill the comprehensive and diverse multi-behavior patterns of users.

Hence, to address the above challenges, in this paper, we propose a new Multi-Behavior recommendation framework with Graph Meta Network (MB-GMN). The goal of MB-GMN is to build a customized meta-learning paradigm upon the multi-behavior relation learning. Specifically, we first propose a meta-knowledge learner for encoding behavior heterogeneity with the consideration of personalized user-specific interactions. After that, a meta graph neural network is developed to capture diverse multi-behavior patterns with high-order connectivity over the user-item interaction graph. With the graph-structured embedding propagation paradigm as our multi-behavior pattern modeling component, the behavior heterogeneity and diversity can be jointly preserved in our learned latent user/item representations. Finally, we introduce a meta prediction network to capture the complex cross-type behavior dependency under a multi-task learning framework. By doing so, multi-typed user behaviors are not only used to modulate the model parameters of graph neural models, but also guide the prediction phase with the injection of supervising signals.

In summary, our major contributions are three-fold:

- To the best of our knowledge, we are the first to study the multi-behavior recommendation problem with the joint modeling behavior heterogeneity and interaction diversity.
- We develop a new framework, MB-GMN which incorporates a multi-behavior pattern modeling component with the meta-learning paradigm. The multi-behavioral relations are distilled to customize the graph neural network for capturing the personalized high-order collaborative effects. In addition, a meta prediction network is introduced to supercharge MB-GMN with the learning of cross-type behavior dependency.
- We demonstrate the effectiveness of MB-GMN on three real-world datasets to show that our framework advances the recommendation performance as compared to baselines from various research lines. Additionally, we further perform the ablation study to better understand the model design of MB-GMN.

## 2 PRELIMINARY

We let $U$ and $V$ represent the set of users and items, respectively, *i.e.*, $\mathcal{U} = \{u_1, ..., u_i, ..., u_I\}$ and $\mathcal{V} = \{v_1, ..., v_j, ..., v_J\}$, where $I$ (indexed by $i$) and $J$ (indexed by $j$) represent the numbers of users and items. With the consideration of multiplex user-item interactions, we define a multi-behavior interaction tensor as below:

DEFINITION 1. **Multi-Behavior Interaction Tensor X**. *We define a three-way tensor $X \in \mathbb{R}^{I \times J \times K}$ to reflect the multi-typed interaction (e.g., click, tag-as-favorite, purchase) between user ($u_i \in \mathcal{U}$) and item ($v_j \in \mathcal{V}$). Here, $K$ denotes the number of behavior types. Specifically, the individual element $x_{i,j}^k \in X$ is set as 1 if user $u_i$ interacts with item $v_j$ under the $k$-th behavior type, and $x_{i,j}^k = 0$ otherwise.*

In the multi-behavior recommendation scenario, one type of user-item interaction will be treated as *target behavior* (*e.g.*, purchase). Other behaviors are referred to as *context behaviors* (*e.g.*, click, tag-as-favorite, add-to-cart) for providing insightful knowledge in assisting the target behavior prediction. Based on the aforementioned definitions, we formally state our studied problem as:

- **Input**: the observed multi-behavior interaction tensor $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ between users $\mathcal{U}$ and items $\mathcal{V}$ across $K$ behavior types.

- **Output**: the predictive function which estimates the likelihood of user $u_i$ adopts the item $v_j$ with the target behavior type.

## 3 METHODOLOGY

We now elaborate the proposed MB-GMN, the model flow of which is illustrated in Figure 1. MB-GMN is composed of two key components: i) multi-behavior pattern encoding, a meta-knowledge learner that captures the personalized multi-behavior characteristics; ii) cross-type behavior dependency modeling, a transfer learning paradigm which learns a well-customized prediction network by transferring knowledge across different behavior types.

### 3.1 Multi-Behavior Pattern Modeling

Unlike previous multi-behavior recommendation models [9, 21, 35] which parameterize multi-typed interaction patterns of different individuals into the same representation space, we aim to capture the personalized multi-behavior characteristics corresponding to diverse user preferences. Towards this end, we propose an integrative neural architecture with a meta-knowledge learner and meta graph neural network. This new framework jointly captures the user-item relation heterogeneity under multi-typed behavior context, as well as the mutual relationships between different behaviors, with respect to their user-specific patterns.
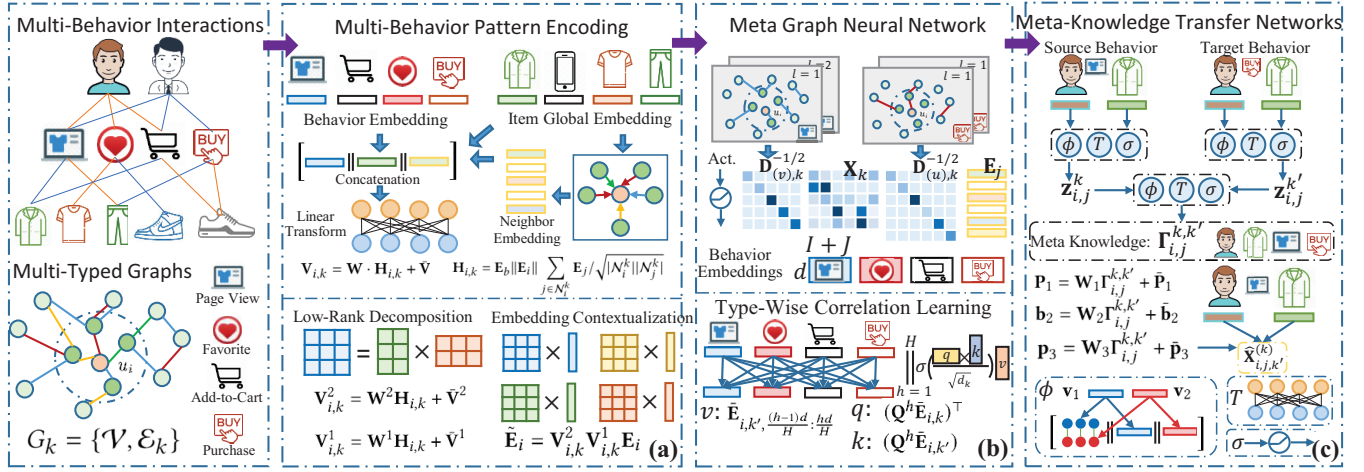
Figure 1: The model architecture of MB-GMN. (a) Multi-behavior pattern modeling with meta-knowledge learner for behavior heterogeneity; (b) Meta graph neural network which preserves the behavior semantics with high-order connectivity; (c) Meta-knowledge transfer networks that customize the parameter of prediction layer to capture cross-type behavior dependency.

*3.1.1* **Meta-knowledge Learner for Behavior Heterogeneity.** In this component, we aim to learn the meta-knowledge from user-specific multi-behavior interactive patterns over different items. We then utilize the learned meta-knowledge to generate the transformation weights for injecting the type-specific behaviour semantics into the initial embeddings. Specifically, given the initialized ID-corresponding embeddings $\mathbf{E}_i \in \mathbb{R}^d$ and $\mathbf{E}_j \in \mathbb{R}^d$ for user $u_i$ and item $v_j$, respectively, we exert the meta network to learn a personalized context projection layer with latent transformations:

$$\mathbf{V}_{i,k} = \mathbf{W} \cdot \mathbf{H}_{i,k} + \bar{\mathbf{V}}; \quad \mathbf{H}_{i,k} = \mathbf{E}_b \| \mathbf{E}_i \| \sum_{j \in \mathcal{N}_i^k} \mathbf{E}_j / \sqrt{|\mathcal{N}_i^k||\mathcal{N}_j^k|} \quad (1)$$

where $\|$ denotes the concatenation operation. $\mathbf{V}_{i,k} \in \mathbb{R}^{d \times d}$ is the learned customized parametric matrix for user $u_i$, which injects the type-specific behavior context into personalized user representation. $\mathbf{W} \in \mathbb{R}^{d \times d \times d}$ and $\bar{\mathbf{V}} \in \mathbb{R}^{d \times d}$ represent the transformation parameters of the meta network. $\mathbf{H}_{i,k} \in \mathbb{R}^d$ is the context-aware embedding which preserves type-specific user interaction behavior patterns. Here, we adopt $\sqrt{|\mathcal{N}_i^k||\mathcal{N}_j^k|}$ as the normalization term for embedding aggregation. In general, this proposed meta-knowledge learner takes the user-specific behavior characteristics as input, *i.e.*, $\mathbf{E}_i$, $\mathbf{E}_b$, as well as personalized interactions over his/her connected items $\sum_{j \in \mathcal{N}_i^k} \mathbf{E}_j$ under the $k$-th behavior type, where $\mathcal{N}_i^k$ denotes the set of interacted items of user $u_i$. The model output is the parametric matrix $\mathbf{V}_{i,k} \in \mathbb{R}^{d \times d}$ which is used to generate weights of graph neural network for handling behavior heterogeneity.

*3.1.2* **Low-rank Transformation Decomposition.** While we can express the customized context representations for user-specific multi-behaviors, it is challenging to extract such meta-knowledge in an efficient way. To tackle this challenge, we enhance our meta-knowledge learner with the low-rank decomposition technique for transformation operation. In particular, the learning phase over the meta-knowledge transformation $\mathbf{V}_{i,k} \in \mathbb{R}^{d \times d}$ can be decomposed

into two low-rank projections as below:

$$\tilde{\mathbf{E}}_i = \mathbf{V}_{i,k}^2 \mathbf{V}_{i,k}^1 \mathbf{E}_i; \quad \mathbf{V}_{i,k}^1 = \mathbf{W}^1 \mathbf{H}_{i,k} + \bar{\mathbf{V}}^1; \quad \mathbf{V}_{i,k}^2 = \mathbf{W}^2 \mathbf{H}_{i,k} + \bar{\mathbf{V}}^2; \quad (2)$$

where $\tilde{\mathbf{E}}_i \in \mathbb{R}^d$ is the contextualized user embedding of user $u_i$, and $\mathbf{V}_{i,k}^1 \in \mathbb{R}^{d' \times d}$, $\mathbf{V}_{i,k}^2 \in \mathbb{R}^{d \times d'}$ are the decomposed transformation matrices, with $\mathbf{W}^1, \mathbf{W}^2, \bar{\mathbf{V}}^1, \bar{\mathbf{V}}^2$ as trainable parameters. Note that $d'$ is the low-rank state dimensionality which is much smaller than the regular embedding dimension size $d$. Hence, the transformation integration $\mathbf{V}_{i,k} = \mathbf{V}_{i,k}^2 \mathbf{V}_{i,k}^1$ is restrict to be low-rank. For the item side, the corresponding embeddings $\mathbf{E}_j$ is contextualized to yield $\tilde{\mathbf{E}}_j$ under the same meta-knowledge learning framework.

## 3.2 Meta Graph Neural Network

To capture the diverse behavior patterns of different users, we propose a meta graph neural network to enhance latent representation learning, with the preservation of behavior-aware user-item collaborative effects. There are three key sub-modules in the meta graph neural network, *i.e.*, i) Behavior semantic encoding; ii) Behavior mutual dependency learning; iii) High-order multi-behavioral context aggregation. We will describe the technical details of each module in the following subsections.

*3.2.1* **Behavior Semantic Encoding.** In our recommendation scenario with multi-behavioral relations, each individual type of interaction behavior has its own characteristics. Taking the e-commence platform as the example, user's page view behavior is more likely to happen than purchase. Moreover, add-to-cart and purchase behavior may co-occur with high probability. In such cases, we propose to capture the personalized behavior semantic signals during the multi-behavior pattern modeling. Given the learned the meta-knowledge $\tilde{\mathbf{E}}_i$ of each user $u_i$ and $\tilde{\mathbf{E}}_j$ of item $v_j$, we design a behavior-aware message passing strategy to capture user-item graph structure $G_k = \{\mathcal{V}, \mathcal{E}_k\}$, where $\mathcal{V}$ denotes the set of nodes (*i.e.*, users and items), and $\mathcal{E}_k$ represents the set of interaction edges

among $\mathcal{V}$ under the $k$-th behavior type. The behavior context-aware embeddings of user $u_i$ and item $v_j$ are generated as follows:

$$\bar{\mathbf{E}}_{i,k} = \mathbf{E}_i + \sigma\Big(\sum_{j \in \mathcal{N}_i^k} \alpha_{i,j,k} \mathbf{E}_j\Big); \qquad \bar{\mathbf{E}}_{j,k} = \mathbf{E}_j + \sigma\Big(\sum_{i \in \mathcal{N}_j^k} \alpha_{i,j,k} \mathbf{E}_i\Big) \quad (3)$$

where $\bar{\mathbf{E}}_{i,k}, \bar{\mathbf{E}}_{j,k} \in \mathbb{R}^d$ are the smoothed embeddings for user $u_i$ and item $v_j$ under the behavior type of $k$, respectively. We define $\alpha_{i,j,k}$ as the normalization factor for $u_i$-$v_j$ interaction with the $k$-th behavior type, which is formally given as $\alpha_{i,j,k} = 1/\sqrt{|\mathcal{N}_i^k||\mathcal{N}_j^k|}$. We summarize the above embedding propagation procedure over user-item interaction $G_k$ with the matrix format as:

$$\bar{\mathbf{E}}_{(u),k} = \mathbf{E}_{(u)} + \sigma(\mathbf{D}_{(u),k}^{-1/2} \cdot \mathbf{X}_k \cdot \mathbf{D}_{(v),k}^{-1/2} \mathbf{E}_{(v)}) \qquad (4)$$

where $\mathbf{X}_k \in \mathbb{R}^{N \times M}$ denotes the adjacent matrix of graph $G_k$ under the $k$-th behavior type. $\mathbf{D}_{(u),k}^{-1/2}$ and $\mathbf{D}_{(v),k}^{-1/2}$ are the diagonal degree matrices for users and items. Here, $\mathbf{E}_{(u)} \in \mathbb{R}^{N \times d}, \mathbf{E}_{(v)} \in \mathbb{R}^{M \times d}$ corresponds to the user and item embedding tables, respectively. In MB-GMN, we further employ a residual unit to prevent the gradient vanishing issue. Inspired by the convolution optimization scheme for collaborative filtering in LightGCN [15], MB-GMN is not equipped with linear transformations during the message passing. Therefore, with the learned meta-knowledge of behavior heterogeneity, MB-GMN endows the graph-structured information propagation with the interaction modeling in a customized manner under personalized behavior context of individual user.

*3.2.2 Behavior Relation Learning.* In practice, different types of interaction behavior are correlated in a complex way, which brings new challenge for modeling multi-behavioral relationships. To encode the underlying relations across different type of behaviors, we design a multi-behavior relation encoding function, to refine the propagated embeddings by injecting the mutual relational information between different behavior types. The relation encoding function is built upon the attention network:

$$\bar{\mathbf{E}}_{i,K+1} = \sum_{k=1}^{K} \mathbf{H}_{i,k}; \quad \mathbf{H}_{i,k} = \Big\|_{h=1}^{H} \sum_{k'=1}^{K} \beta_{k,k'}^h \cdot \bar{\mathbf{E}}_{i,k',\frac{(h-1)d}{H}:\frac{hd}{H}};$$

$$\beta_{k,k'}^h = \frac{\exp \hat{\beta}_{k,k'}^h}{\sum_{k'=1}^{K} \exp \hat{\beta}_{k,k'}^h}; \qquad \hat{\beta}_{k,k'}^h = \frac{(\mathbf{Q}^h \bar{\mathbf{E}}_{i,k})^\top (\mathbf{Q}^h \bar{\mathbf{E}}_{i,k'})}{\sqrt{d/H}} \qquad (5)$$

where $\tilde{\mathbf{E}}_{i,K+1} \in \mathbb{R}^d$ denotes the global user embedding that takes all behavior types into account. $\mathbf{H}_{i,k} \in \mathbb{R}^d$ is an intermediate representation which encodes the behavior-specific features refined by other behavioral data. $\mathbf{H}_{i,k}$ is calculated using an $H$-head dot-product attention mechanism. Additionally, $\beta_{k,k'}^h \in \mathbb{R}$ is the $h$-th quantitative weight for behavior type $k$ and $k'$, subscript $\frac{(h-1)d}{H} : \frac{hd}{H}$ denotes the $h$-th slice of the vector (each slice is of size $d/H$), and $\mathbf{Q}^h \in \mathbb{R}^{\frac{d}{H} \times d}$ is the $h$-head parametric transformation. To alleviate the overfitting effect and improve the training efficiency, we optimize our attentive encoding function from the following perspectives: 1) we slice the $d$-dimensional $\bar{\mathbf{E}}_{i,k'}$ to $H$ parts as feature vectors corresponding to $H$ heads, without the heavy value transformation; 2) we enable the matrix transformation sharing among the query and key dimensions with the projection parameter $\mathbf{Q}^h$.

*3.2.3 High-order Multi-Behavioral Context Aggregation.* To capture the high-order connectivity of the interaction graph, MB-GMN integrates the behavior semantic encoding with the behavior-wise relational structure learning under the high-order embedding propagation paradigm. In specific, we inject the high-order connectivity into the user embedding $u_i$ as:

$$\bar{\mathbf{E}}_k^{(l+1)} = \begin{cases} \text{Graph-Conv}(\bar{\mathbf{E}}_k^{(l)}), & k = 1, 2, ..., K \\ \text{Att}(\bar{\mathbf{E}}_1^{(l+1)}, ..., \bar{\mathbf{E}}_K^{(l+1)}), & k = K+1 \end{cases}$$

where $\text{Graph-Conv}(\cdot)$ denotes the behavior semantic encoder, and $\text{Att}(\cdot)$ represents the function for behavior dependency learning. By stacking the graph encoders with $L$ times, MB-GMN is aware of the $L$-hop graph-based connection structure for each node. To aggregate the cross-order message embeddings, we integrate the order-specific embeddings and yield the final user and item representations using the operation: $\hat{\mathbf{E}}_k = \sum_{l=0}^{L} \bar{\mathbf{E}}_k^l$ for $k = 1, 2, ..., K+1$.

## 3.3 Meta-Knowledge Transfer Networks

With the design of our meta graph neural network, we realize the information propagation and aggregation between users and items, so as to characterize the personalized multi-behavior patterns vary by users. Now, we move on to the prediction of different types of user-item interactions, based on a meta learning paradigm for customized knowledge transfer. From the meta graph neural network, we could obtain the $(K + 1)$ aggregated behavior representations, *i.e.*, type-specific behavioral patterns are characterized by $K$ different embeddings $\hat{\mathbf{E}}_k$, and a summarized general interaction (without differentiating behavior types) $\hat{\mathbf{E}}_{K+1}$ which preserves the general user interest. In the prediction phase, how to effectively capture the cross-type behavior dependency and transfer the influential knowledge in assisting the forecasting task on the target behavior, remains a significant challenge.

*3.3.1 Meta-knowledge Learner for Behavior Dependency.* In MB-GMN, we propose a meta learning scheme to generate prediction sub-networks with respect to the target user-item pair $(u_i, v_j)$ and behavior type. During our target behavior (*e.g.*, purchase) prediction phase, context behavior information (*e.g.*, click, tag-as-favorite, add-to-cart) not only provides the insightful external knowledge, but also serve as the supervising signals for model optimization. Given user-item pair $u_i$ and $v_j$, our meta prediction network first learns the underlying meta knowledge customized to $u_i$ and $v_j$ under the target behavior type of $k'$ (other context behavior types are indexed by $k$). This process is presented as follows:

$$\Gamma_{i,j}^{k,k'} = \sigma(\mathbf{W}_1 \cdot \phi(\mathbf{Z}_{i,j}^k, \mathbf{Z}_{i,j}^{k'})); \quad \mathbf{Z}_{i,j}^k = \sigma(\mathbf{W}_2 \cdot \phi(\hat{\mathbf{E}}_{i,k}, \hat{\mathbf{E}}_{j,k})) \quad (6)$$

where $\Gamma_{i,j}^{k,k'} \in \mathbb{R}^d$ is the encoded meta knowledge of $u_i$ and $v_j$ with the relation between the target behavior $k'$ and context behaviors $k$ ($k \in (K + 1), k' \in K$). $\mathbf{Z}_{i,j}^k, \mathbf{Z}_{i,j}^{k'} \in \mathbb{R}^d$ are the projected representations of $u_i$ and $v_j$ with the relation between the $k'$-th and the $k$-th behavior type. $\sigma(\cdot)$ is the activation function and $\phi(\cdot)$ is the encoding function to capture the dependent interactive relations between two embeddings. To make use of both the binary relations and the unary bias, MB-GMN proposes to employ a $\phi(\cdot)$ function defined as: $\phi(\mathbf{v}_1, \mathbf{v}_2) = \mathbf{v}_1 \circ \mathbf{v}_2 \| \mathbf{v}_1 \| \mathbf{v}_2$, where $\circ$ denotes element-wise multiplication and $\|$ denotes concatenation operation.

*3.3.2* **Knowledge Transfer Learning Framework.** With the meta-knowledge representation $\Gamma_{i,j}^{k,k'}$, MB-GMN generates three parametric variables for our prediction network as follows:

$$\mathbf{P}_1 = \mathbf{W}_1\Gamma_{i,j}^{k,k'} + \bar{\mathbf{P}}_1; \quad \mathbf{b}_2 = \mathbf{W}_2\Gamma_{i,j}^{k,k'} + \bar{\mathbf{b}}_2; \quad \mathbf{p}_3 = \mathbf{W}_3\Gamma_{i,j}^{k,k'} + \bar{\mathbf{p}}_3 \quad (7)$$

where $\mathbf{P}_1 \in \mathbb{R}^{d \times 3d}, \mathbf{b}_2, \mathbf{p}_3 \in \mathbb{R}^d$ are the learned parameters for $u_i$, $v_j$, source task $k$ and target task $k'$. $\mathbf{W}_*$ are the parameters of the meta network, and $\bar{\mathbf{P}}_1, \bar{\mathbf{b}}_2, \bar{\mathbf{p}}_3$ are the bias parameters for $\mathbf{P}_1, \mathbf{b}_2$ and $\mathbf{p}_3$. Finally, MB-GMN makes predictions on $u_i, v_j$ under the target behavior $k'$, using source data of context behavior $k$, by:

$$\hat{\mathbf{X}}_{i,j,k'}^{(k)} = \eta^\top \mathbf{p}_3; \quad \eta = \sigma(\mathbf{P}_1 \cdot \phi(\hat{\mathbf{E}}_{i,k}, \hat{\mathbf{E}}_{j,k}) + \mathbf{b}_2) \quad (8)$$

where $\hat{\mathbf{X}}_{i,j,k'}^{(k)} \in \mathbb{R}$ is the predicted likelihood of $u_i$ interacting with $v_j$ by behavior $k'$, given the interaction data $k$ as features. $\eta \in \mathbb{R}^d$ is an intermediate feature vector. For the most concerned target behavior $k'$ (*e.g.* purchase), we use $\hat{\mathbf{X}}_{i,j,k'}^{(K+1)}$ as the final prediction.

## 3.4 Multi-task Learning Framework

We train MB-GMN by optimizing the prediction objective for each pair of source and target behavior. For user $u_i$ and target behavior $k'$, MB-GMN samples $S$ positive interactions and $S$ non-interacted items as the negative samples. During the training phase, we adopt the Adam algorithm to optimize the following defined objective:

$$\mathcal{L} = \sum_{i=1}^N \sum_{k=1}^{K+1} \sum_{k'=1}^K \sum_{s=1}^S \max(0, 1 - \hat{\mathbf{X}}_{i,p_s,k'}^{(k)} + \hat{\mathbf{X}}_{i,n_s,k'}^{(k)}) + \lambda\|\Theta\|_{\mathrm{F}}^2 \quad (9)$$

where $k$ refers to the source behavior and $k'$ refers to the target behavior, $p_s$ and $n_s$ are the $s$-th positive and negative samples, respectively. The first term is the marginal pair-wise loss, and the last term denotes the weight-decay regularization with weight $\lambda$.

*3.4.1* **Model Complexity Analysis.** The time cost of MB-GMN mainly comes from the following aspects. i) In the multi-behavior pattern modeling, MB-GMN adopts lightweight graph convolutional architecture which costs only $O(L \times K \times d \times |\mathcal{E}|)$ across $L$ layers, $K$ behavior types, $d$ latent factors and $|\mathcal{E}|$ edges. The behavior relation learning costs extra $O(L \times K \times d \times (K + d) \times (N + M))$. As $O(d \times |\mathcal{E}|)$ is comparable with $O((K + d) \times (N + M))$ in our case, the complexity doesn't increase. The contextualization meta network costs $O(K \times (N + M) \times d^2 \times d')$ for parameter learning, and $O(K \times (N + M) \times d \times d')$ for embedding transformation, which is very close to the complexity of GCN networks. The prediction meta networks costs minor $O(S \times d^2)$ computations for each user. In conclusion, our MB-GMN could achieve comparable model time complexity with some graph convolution-based models.

## 4 EVALUATION

This section performs experiments on different datasets to evaluate the performance of our *MB-GMN* framework by making comparison with various state-of-the-art recommendation techniques. In particular, we aim to answer the following research questions:

- **RQ1**: How does *MB-GMN* perform when competing with various recommendation baselines on different datasets?

**Table 1: Statistics of experimented datasets**

| Dataset | User # | Item # | Interaction # | Interactive Behavior Type |
|---|---|---|---|---|
| Taobao-Data | 147894 | 99037 | 7658926 | {Page View, Favorite, Cart, Purchase} |
| Beibei-Data | 21716 | 7977 | 3338068 | {Page View, Cart, Purchase} |
| IJCAI-Contest | 423423 | 874328 | 36203512 | {Page View, Favorite, Cart, Purchase} |

- **RQ2**: How do the designed sub-modules in our *MB-GMN* framework affect the recommendation performance?

- **RQ3**: How does *MB-GMN* work for making recommendation to users with the consideration of different types of behaviors?

- **RQ4**: How does *MB-GMN* perform when handling behavior data with different interaction sparsity degrees?

- **RQ5**: How do different configurations of key hyperparameters impact the performance of *MB-GMN* framework?

- **RQ6**: How is the model explainability of our *MB-GMN*? What relational patterns between different types of behaviors could *MB-GMN* capture in assisting the final recommendation task?

### 4.1 Experiment Settings

*4.1.1* **Dataset.** To evaluate the effectiveness of our model, we conduct experiments on three datasets from real-world platforms. We describe the data details as below.

**Taobao Data**. There are four types of user-item relations contained in this data from Taobao–one of the largest e-commerce platform in China, *i.e.*, *page view*, *add-to-cart*, *tag-as-favorite* and *purchase*.

**Beibei Data**. This data is collected from Beibei which is one of the largest infant product online retailing site in China. It involves three types of user-item interactions, including *page view*, *add-to-cart* and *purchase*, to generate the multi-behavior interaction tensor **X**.

**IJCAI Data**. This data is released by IJCAI competition for user activity modeling from an online business-to-consumer e-commerce platform. Four types of behaviors are included in this dataset, *i.e.*, *page view*, *add-to-cart*, *tag-as-favorite* and *purchasing*.

In those three datasets, we regard users' purchases as the target behaviors, because purchases are good indicators for Gross Merchandise Volume which indicates the total volume of sales in real-life online retailing sites [13, 32]. We further summarize the detailed information of each experimented dataset in Table 1.

*4.1.2* **Evaluation Protocols.** For performance evaluations, we use two representative metrics: *Normalized Discounted Cumulative Gain (NDCG@N)* and *Hit Ratio (HR@N)* which have been widely adopted in top-N recommendation tasks [2, 29]. Following the same experimental settings in [16, 17], the leave-one-out evaluation is leveraged for training and test set partition. We generate the test data set by including the last interactive item and consider the rest of data for training. For efficient and fair model evaluation, we pair each positive item instance with 99 randomly sampled non-interactive items for each user, which shares similar settings in [22, 26].

*4.1.3* **Methods for Comparison.** To comprehensively evaluate the performance, we compare our *MB-GMN* against various baselines from different research lines, which are elaborated as below: **Conventional Matrix Factorization Approach**:

- **BiasMF** [23]: this approach considers the bias information from users and items using the matrix factorization model.

**Neural Collaborative Filtering Methods**:

- **DMF** [37]: this model considers the explicit interaction and non-preference implicit feedback into a deep matrix factorization model for user/item embedding mapping.
- **NCF** [16]: we consider three variants of NCF with the configuration of different interaction encoders: *i.e.*, concatenated element-wise-product branch (*i.e.*, NCF-N), Multilayer perceptron (*i.e.*, NCF-M), and the fixed element-wise product (*i.e.*, NCF-G).

**Autoencoder-based Collaborative Filtering Models**:

- **CDAE** [34]: it applies the denoising autoencoders to train a neural network with the data reconstruction objective in an adaptive way.
- **AutoRec** [25]: it stacks multi-layer autoencoder to map user-item interactions into latent low-dimensional representations.

**Neural Auto-regressive Recommendation Models**:

- **NADE** [42]: it is a feed-forward autoregressive architecture for collaborative filtering with the parameter sharing technique.
- **CF-UIcA** [8]: it explicitly captures the relations between items and users with a neural co-autoregressive model.

**Graph Neural Networks for Recommender Systems**:

- **ST-GCN** [40]: this method is a convolution-based graph neural model which generates user embeddings based on an encoder-decoder framework.
- **NGCF** [29]: it is a message passing architecture for information aggregation over the user-item interaction graph, to exploit high-order connection relationships.

**Recommendation with Multi-Behavior Relations**.

- **NMTR** [9]: it is a multi-task recommendation framework for performing the cascade prediction of different types of behaviors.
- **DIPN** [13]: it aims to predict the purchase intent of users by combing different user interactive patterns based on attention.
- **NGCF$_M$** [29]: we enhance the NGCF model to inject the multi-behavioral graph relations under a graph neural network.
- **MATN** [35]: it differentiates the relations between user and item with the integration of the attention network and memory units.
- **MBGCN** [21]: it designs a unified graph to represent the multi-behavior of users and uses graph convolutional network to perform behavior-aware embedding propagation.

*4.1.4* **Parameter Settings.** We implement the proposed MB-GMN model using TensorFlow. The parameters are learned using the Adam optimizer, with the learning rate $1e^{-3}$ and decay rate 0.96. The regularization weight $\lambda$ is selected from {0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001}. The training batch size is selected from {32, 128, 256, 512}. By default, MB-GMN is configured with the hidden state dimensionality of 16, low-rank dimensionality of 4, 2 attention heads. The number of message passing layers in our meta graph neural network is searched from {1,2,3}. The experiments for most of neural network baselines are performed based on their released codes or presented settings. For NCF and NMTR, we sample the positive and negative instances with the range of 1 : 1 to 1 : 4 to fit their model optimization with point-wise loss.

## 4.2 Performance Validation (RQ1)

We evaluate the performance of all baseline methods in predicting the target user interactive behavior on different datasets, and summarize the following observations:

Table 2 shows the performance of top-*N* item recommendation under the target behavior type and "Imp" indicates the relatively improvement ratio between our *MB-GMN* and each baseline. From the presented results, we observe that the recommendation performance could be obviously improved by *MB-GMN*. Such performance gap can be attributed to the effective modeling of personalized multi-behavior patterns, which results in customized user/item representations under a meta learning paradigm.

We can also observe that *MB-GMN* consistently obtains better performance than several baselines which models multi-typed user behaviors for recommendations. For example, the evaluation results shed light on the limitation of NMTR method, by merely modeling the singular dimensional cascading correlations between multi-type interactions. The dual-stage attention mechanisms (DIPN and MATN) which aggregate different types of behavioral patterns through weighted summation, cannot comprehensively capture the complex inter-dependencies across different types of user behavior. Additionally, the large performance gap between *MB-GMN* and compared multi-behavior graph neural frameworks (*i.e.*, NGCF$_M$ & MBGCN), justifies the advantage of our method for performing customized multi-behavior dependency modeling. In contrast, these GNN-based baselines aggregate the behavior-aware patterns under a parameter sharing message passing architecture, which overlook the unique characteristics of user behavior patterns.

Among various compared methods, we can observe that the injection of multi-behavior information into the recommendation frameworks (*i.e.*, NMTR, DIPN, NGCF$_M$, MATN, MBGCN) could boost the performance, as compared to other baselines which do not differentiate user-item interactions. This observation confirms the benefits of exploring multi-behavior patterns for modeling collaborative effects in recommendation. Furthermore, GNN-based models are superior to other baselines, which suggests that high-order relations pertinent to users and items are derived with stacking message passing layers.

We also evaluate the performance of *MB-GMN* and several representative baselines while varying the value of *N* in terms of HR and NDCG. Table 3 shows the evaluation results on Beibei dataset. We could observe that *MB-GMN* always achieves the best performance with different top-*N* positions. This observation indicates the consistent superiority of our developed framework as compared to other baselines, in giving correctly interacted items with higher probability.

## 4.3 Model Ablation Study (RQ2)

To evaluate the rationality of design sub-modules in our *MB-GMN* framework, we consider five model variants as follows:

- ***w/o LowR***: We do not integrate the low-rank transformation decomposition in our meta-knowledge learner for encoding behavior heterogeneity. The regular learned meta-transformer $\mathbf{V}_{i,k} \in \mathbb{R}^{d \times d}$ is directly used for context embedding.

**Table 2: Overall model performance on Beibei, Taobao and IJCAI data, with the metrics of *HR@N* and *NDCG@N* ($N$ = 10).**

| Data | Metric | BiasMF | DMF | NCF-M | NCF-G | NCF-N | AutoRec | CDAE | NADE | CF-UIcA | ST-GCN | NGCF | NMTR | DIPN | NGCF$_M$ | MBGCN | MATN | *MB-GMN* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Beibei | HR | 0.588 | 0.597 | 0.595 | 0.594 | 0.601 | 0.607 | 0.608 | 0.608 | 0.610 | 0.609 | 0.611 | 0.613 | 0.631 | 0.634 | 0.642 | 0.626 | **0.691** |
| | Imprv | 17.52% | 15.75% | 16.13% | 16.33% | 14.98% | 13.84% | 13.65% | 13.65% | 13.28% | 13.46% | 13.09% | 12.72% | 9.51% | 8.99% | 7.63% | 10.38% | – |
| | NDCG | 0.333 | 0.336 | 0.332 | 0.338 | 0.336 | 0.341 | 0.341 | 0.343 | 0.346 | 0.343 | 0.375 | 0.349 | 0.394 | 0.372 | 0.376 | 0.385 | **0.410** |
| | Imprv | 23.12% | 22.02% | 23.49% | 21.30% | 22.02% | 20.23% | 20.23% | 19.53% | 18.50% | 19.53% | 9.33% | 17.48% | 4.06% | 10.22% | 9.04% | 6.49% | – |
| Taobao | HR | 0.262 | 0.305 | 0.319 | 0.290 | 0.325 | 0.313 | 0.329 | 0.317 | 0.332 | 0.347 | 0.302 | 0.332 | 0.317 | 0.374 | 0.369 | 0.354 | **0.491** |
| | Imprv | 87.40% | 60.98% | 53.92% | 69.31% | 51.08% | 56.87% | 49.24% | 54.89% | 47.89% | 41.50% | 62.58% | 47.89% | 54.89% | 31.28% | 33.06% | 38.70% | – |
| | NDCG | 0.153 | 0.189 | 0.191 | 0.167 | 0.190 | 0.190 | 0.191 | 0.196 | 0.198 | 0.206 | 0.185 | 0.179 | 0.178 | 0.221 | 0.222 | 0.209 | **0.300** |
| | Imprv | 96.08% | 58.73% | 57.07% | 79.64% | 49.25% | 57.89% | 53.06% | 57.07% | 51.52% | 45.63% | 62.16% | 67.60% | 68.54% | 35.75% | 35.14% | 43.54% | – |
| IJCAI | HR | 0.285 | 0.392 | 0.449 | 0.292 | 0.459 | 0.448 | 0.455 | 0.469 | 0.429 | 0.452 | 0.461 | 0.481 | 0.475 | 0.481 | 0.463 | 0.489 | **0.532** |
| | Imprv | 86.67% | 35.71% | 18.49% | 82.19% | 15.90% | 18.75% | 16.92% | 13.43% | 24.01% | 17.70% | 15.40% | 10.60% | 12.00% | 10.60% | 14.90% | 8.79% | – |
| | NDCG | 0.185 | 0.250 | 0.284 | 0.188 | 0.294 | 0.287 | 0.288 | 0.304 | 0.260 | 0.285 | 0.292 | 0.304 | 0.296 | 0.307 | 0.277 | 0.309 | **0.345** |
| | Imprv | 86.49% | 38.00% | 21.48% | 83.51% | 17.35% | 20.21% | 19.79% | 13.49% | 32.69% | 21.05% | 18.15% | 13.49% | 16.55% | 12.38% | 24.55% | 11.65% | – |

**Table 3: Rank-based prediction accuracy via varying Top-*N* value measured by *HR@N* and *NDCG@N* on Beibei data.**

| Model | @1 | | @3 | | @5 | | @7 | |
|---|---|---|---|---|---|---|---|---|
| | HR | NDCG | HR | NDCG | HR | NDCG | HR | NDCG |
| BiasMF | 0.118 | 0.118 | 0.310 | 0.228 | 0.453 | 0.287 | 0.537 | 0.316 |
| NCF-N | 0.123 | 0.123 | 0.317 | 0.232 | 0.447 | 0.283 | 0.530 | 0.315 |
| NGCF-M | 0.167 | 0.167 | 0.376 | 0.286 | 0.496 | 0.337 | 0.569 | 0.361 |
| MBGCN | 0.167 | 0.167 | 0.374 | 0.284 | 0.498 | 0.337 | 0.579 | 0.363 |
| AutoRec | 0.128 | 0.128 | 0.321 | 0.236 | 0.456 | 0.291 | 0.540 | 0.322 |
| MATN | 0.184 | 0.184 | 0.361 | 0.286 | 0.467 | 0.330 | 0.543 | 0.356 |
| *MB-GMN* | **0.183** | **0.183** | **0.401** | **0.306** | **0.527** | **0.359** | **0.608** | **0.389** |

**Table 4: Ablation study on key components of MB-GMN.**

| Data | Beibei | | Taobao | | IJCAI Contest | |
|---|---|---|---|---|---|---|
| Metrics | HR | NDCG | HR | NDCG | HR | NDCG |
| w/o LowR | 0.6825 | 0.4083 | 0.4385 | 0.2656 | 0.4931 | 0.3195 |
| w/o MFeat | 0.6813 | 0.4049 | 0.3878 | 0.2315 | 0.4865 | 0.3138 |
| w/o MTask | 0.6068 | 0.3401 | 0.4577 | 0.2782 | 0.5301 | 0.3392 |
| w/o MetaC | 0.6791 | 0.4046 | 0.4647 | 0.2852 | 0.5026 | 0.3199 |
| w/o MetaP | 0.6605 | 0.4036 | 0.4868 | 0.2968 | 0.5302 | 0.3399 |
| *MB-GMN* | **0.6907** | **0.4103** | **0.4906** | **0.2997** | **0.5319** | **0.3447** |

- ***w/o MFeat***: The behavior relation learning sub-module is removed in the stage of multi-behavior pattern modeling.

- ***w/o MTask***: We do not perform the multi-task learning in our meta prediction network with the joint model optimization.

- ***w/o MetaC***: We do not leverage the designed meta-knowledge learner in encoding the behavior heterogeneous. Instead, we pre-define different transformations for multi-typed behaviors.

- ***w/o MetaP***: We do not rely on the meta-knowledge learner to learn parameters of our prediction layer. The pre-defined parameter space is adopted for discriminating representation space.

The ablation study results are shown in Table 4. From the evaluation results, we draw the following conclusions:

- The performance gap between *MB-GMN* and the w/o LowR variant indicates the advantage of our designed low-rank transformation decomposition scheme in alleviating the overfitting issue during the heavy transformation operations, by serving as an efficient regularization term in our meta-knowledge learner.

- The behavior relation learning shows positive effects for our meta graph neural network during the customized message passing procedure. This observation demonstrates the rationality of our utilized attention layer under multiple representation subspaces, in capturing pair-wise correlations among various interaction behavior types.

- The evaluation results shed light on the limitation of single-task learning paradigm without the consideration of cross-type behavior dependency in the prediction phase. This suggests that context behavior relations can not only provide the insightful external knowledge during the explicitly multi-behavior pattern aggregation, but also serve as the supervising signals for model optimization under multi-task learning paradigm.

- *MB-GMN* outperforms both w/o MetaC and w/o MetaP, due to their simplified design without the incorporation of meta-knowledge learner. The identified meta-knowledge is very important for performing user-specific behavior modeling through a customized meta-learning framework.

## 4.4 Effect of Context Behaviors (RQ3)

We further perform the ablation study to show the effectiveness of incorporating different context behaviors into our *MB-GMN* framework. The evaluation results are presented in Figure 2. Here, "-pv", "-fav", "-cart" represents our *MB-GMN* without the incorporation of individual "page view", "tag-as-favorite", "add-to-cart" behavior, respectively. "+buy" is the variant which only relies on the target purchase behaviors themselves to make prediction. From results, we can observe that our framework, which collectively captures user preferences with all types of context behaviors, achieves the best performance as compared to other variants. This demonstrates the effectiveness of learning the interaction heterogeneity to reflect comprehensive multi-behavior patterns.

## 4.5 Performance v.s. Sparsity Degrees (RQ4)

Since interaction behavioral patterns vary by users and data sparsity phenomenon is ubiquitous in many real-world recommendation scenarios, we evaluate how is the performance of our new recommendation framework when handling user behavior data with different sparsity degrees. To be specific, x-axis in Figure 3 represents the number of user interactions, and y-axis indicates the recommendation accuracy (measured by NDCG@10 and HR@10).

From the results, we can observe the consistent performance gain of our *MB-GMN* framework over other compared methods, which suggests the model robustness of *MB-GMN* in capturing user's preference under different data sparsity degrees. Furthermore, we can notice that it is important to capture the multi-behavior patterns, as *MB-GMN* and MBGCN achieve better performance as
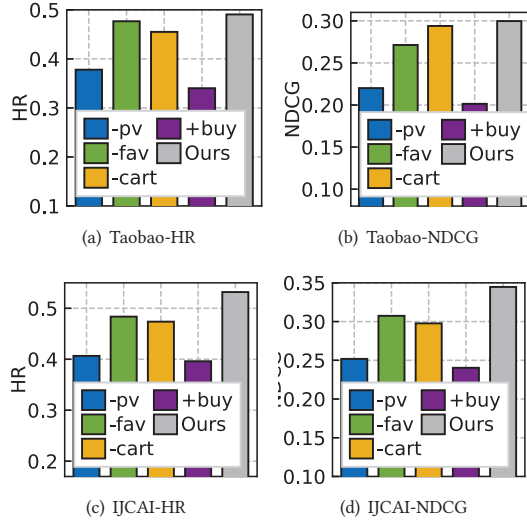
Figure 2: Impact study of different types of context behavior.

compared other types of baselines. It is reasonable that incorporating context behaviors (*e.g.*, page view, tag-as-favorite) results in better performance by considering multi-view user preferences. This performance improvement also shows that the incorporation of multi-behaviour information of users can help to alleviate the negative effects of data sparsity in encoding user preference.
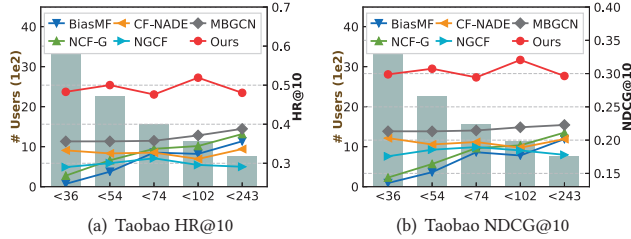


Figure 3: Performance of MB-GMN and baseline methods *w.r.t* different data sparsity degrees on Taobao data.

## 4.6 Hyperparameter Study (RQ5)

To show the effect of different parameter settings, we perform experiments to evaluate the performance of our developed *MB-GMN* framework with different configurations of key hyperparameters, namely, the hidden state dimensionality $d$, low-rank decomposition dimension $d'$, and the depth $L$ of our graph neural network. We show the results in Figure 4. Considering that accuracy vary by different datasets in terms of HR and NDCG metric ranges, the y-axis of Figure 4 represents the estimated performance decrease/increase percentage of each parameter-specific performance as compared to the best performance with the optimal hyperparameter settings. The discussions are summarized as follows.

- **Hidden State Dimensionality:** $d$. In our framework, the latent state dimensionality $d$ is searched from 4 to 32. We can observe that with the increasing of embedding dimension from 4 to 16, the
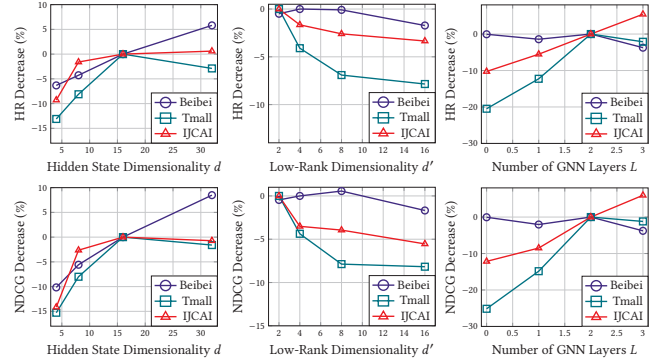


Figure 4: Hyperparameter study of the MB-GMN.

recommendation performance improves due to the stronger representation feature space. However, a larger embedding dimensionality does not always bring stronger model representation ability in learning multi-behavioral relations for recommendations. This is caused by the model overftting.

- **Low-rank Decomposition Dimension** $d'$. We can notice that smaller value of $d'$ in low-rank transformation decomposition is more beneficial for learning meta-knowledge from behavior heterogeneity. The reason lies in that the relatively small value of $d' \leq 4$ could reduce the computational cost and improve the model robustness. This observation again confirms the rationality of our design low-rank decomposition scheme in learning meta-knowledge from multi-behavior user data.

- **Depth of Graph Neural Paradigm** $L$. By stacking two graph neural layers, *MB-GMN*-2 performs better than *MB-GMN*-1, which suggests the positive effect of high-order connectivity injection. However, with the increasing of depth of graph model, the performance starts to deteriorate. This is because stacking more embedding propagation layers may involve noise signals for modeling collaborative effects between users and items, which leads to the over-smoothing [6].

## 4.7 Model Interpretation Study (RQ6)

We further perform case studies with sampled case examples to show the interpretation capability of *MB-GMN* by capturing the cross-type behavior dependent structures. In the experiments, we visualize the loss value during our training phase after the model convergence, so as to reflect the helpfulness of individual context behavior in predicting the target behavior. In our joint learning framework, each type of user-item interaction behavior could be considered as context source behavior and target behavior. In such cases, we generate a $4 \times 4$ weight matrix from six sampled users (*e.g.*, $u_{245}$, $u_{308}$) corresponding to page view, tag-as-favorite, add-to-cart and purchase (as shown in Figure 5). In general, we can observe that page view behaviors contribute more on predicting other types of behaviors in most cases. The overall helpfulness evaluation results across users are also illustrated with the heatmap matrix on the right side. Similar results can be observed. The potential reason for this observation lies in the larger amount of page view activities of users compared with other types of user interactions. General speaking, all above observations could show the interpretation ability of our
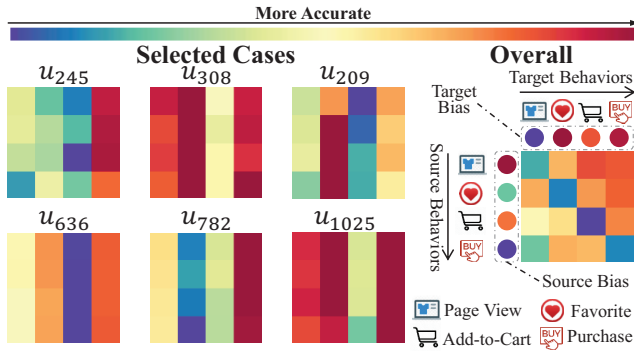
**Figure 5: Case study on the explainability of *MB-GMN*. Source bias represents the helpfulness of individual behavior in serving as context behavior in providing auxiliary information. Target bias represents the usefulness for predicting this target behavior based on the context behavior representation. The type-wise behavior dependency is reflected with heatmap matrix, *i.e.*, six heatmap $4 \times 4$ matrices corresponding to six extracted users, one heatmap $4 \times 4$ matrix presents the overall dependencies across different types of behaviors from all users.**

*MB-GMN* architecture in capturing complex dependency across different types of user behaviors and characterize the underlying helpfulness of context behavior during the prediction.

## 5 RELATED WORK

### 5.1 Collaborative Filtering With Deep Learning

In recent years, augmenting collaborative filtering techniques with deep neural networks has become increasingly popular, due to the strong capability of neural architectures in learning complex interactive patterns between users and items [7, 38]. There exist various types of deep learning paradigms, with structures suited to collaborative filtering architectures. Specifically, NCF [16] proposed to enhance the feature interaction between projected user and item embeddings with Multi-Layer Perceptron component, so as to address the limitation of inner-product operation in the recommendation phase. DMF [37] extended the matrix factorization with neural networks to map users and items into a common latent space. Autoencoder and its model variants have also been utilized for collaborative filtering tasks with the reconstruction-based encoder-decoder learning over user-item interactions [25, 34]. Different from the above deep collaborative filtering models which merely consider single-typed interactive behavior of users, our MB-GMN explores the users' multi-behavioral interactions and effectively incorporates the cross-behavior inter-dependent knowledge into the recommender systems, to boost the performance.

### 5.2 Multi-Relation Recommendation Models

Many methods were proposed to enhance the representation capability of recommender systems by characterizing various relations from the side information of users and items [18, 24]. Let us discuss some recent literature: a research line of recommendation models proposed to incorporate online social relations of users to assist the user representation, by considering the influence between different

users [5, 20, 33]. Furthermore, knowledge graph information has also become an important data source to enhance existing recommendation frameworks. These knowledge-aware recommendation models made use of connections between different entities to supplement the interaction learning between users and items [27, 28, 43]. For instance, a knowledge-aware attentive reasoning scheme is proposed to aggregate information from user behaviors and relational paths in the generated knowledge graphs [43].

### 5.3 Graph Neural Networks for Recommendation

Recent years have witnessed the effectiveness of graph neural networks (GNNs) for mining various relational data [4, 39]. In general, the key idea of GNN models is to perform graph-structured learning with message passing architectures for aggregating feature information from neighborhood nodes. For example, Graph Convolutional Network (GCN) [11], GraphSAGE [14] and Graph Attention Network (GAT) [30] are several representative GNN models which aggregate feature information from neighboring vertices, by utilizing convolutional operation, or attentional mechanism. GNNs have also been utilized to augment the user-item interaction modeling based on graph structures. For example, NGCF [29] and ST-GCN [40] are message passing recommendation methods which design aggregation encoder with graph convolutional network. Zheng *et al.* [41] learns price-aware purchase intent of users with a GCN-based architecture. Inspired by the above research work, we propose a new neural approach MB-GMN within the broader graph structure for multi-behavior recommendation, to tackle the technical challenges for dealing with the interaction diversity.

## 6 CONCLUSION

This work develops a novel multi-behavior enhanced recommendation framework with graph meta network for learning interaction heterogeneity and diversity. Our framework explicitly incorporates the inter-correlations of multiple types of user behavior into the collaborative filtering architecture, which are usually neglected in most existing recommendation methods. In addition, our proposed framework incorporates the meta-knowledge network into a meta-learning paradigm, which endows the multi-behavior pattern modeling in a customized manner. Extensive experiments on three large-scale E-commerce datasets verify the effectiveness of MB-GMN. Several promising direction of future work are: First, we plan to extend the developed MB-GMN framework by incorporating side context information into the modeling process of user's preference, such as user's profile information and items' text descriptions. Second, a time-sensitive model is deserved to be investigated in order to handle the new arriving data streaming of users' behavior in a timely manner, which benefits the real-time recommendation.

# REFERENCES

[1] Kristen M Altenburger and Daniel E Ho. 2019. Is Yelp Actually Cleaning Up the Restaurant Industry? A Re-Analysis on the Relative Usefulness of Consumer Reviews. In *WWW*. 2543–2550.

[2] Ting Bai, Lixin Zou, Wayne Xin Zhao, Pan Du, Weidong Liu, Jian-Yun Nie, and Ji-Rong Wen. 2019. Ctrec: A long-short demands evolution model for continuous-time recommendation. In *SIGIR*. 675–684.

[3] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation learning for attributed multiplex heterogeneous network. In *KDD*. 1358–1368.

[4] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *NIPS*. 4869–4880.

[5] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *SIGIR*. 225–234.

[6] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI*, Vol. 34. 3438–3445.

[7] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten De Rijke. 2019. Joint neural collaborative filtering for recommender systems. *TOIS* 37, 4 (2019), 1–30.

[8] Chao Du, Chongxuan Li, Yin Zheng, Jun Zhu, and Bo Zhang. 2018. Collaborative filtering with user-item co-autoregressive models. In *AAAI*.

[9] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. 2019. Neural multi-task recommendation from multi-behavior data. In *ICDE*. IEEE, 1554–1557.

[10] Chen Gao, Chao Huang, Yue Yu, Huandong Wang, Yong Li, and Depeng Jin. 2019. Privacy-preserving cross-domain location recommendation. *Ubicomp* 3, 1 (2019), 1–21.

[11] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. 2018. Large-scale learnable graph convolutional networks. In *KDD*. 1416–1424.

[12] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. 2020. Hierarchical User Profiling for E-commerce Recommender Systems. In *WSDM*. 223–231.

[13] Long Guo, Lifeng Hua, Rongfei Jia, Binqiang Zhao, Xiaobo Wang, and Bin Cui. 2019. Buying or Browsing?: Predicting Real-time Purchasing Intent using Attention-based Deep Network with Multiple Behavior. In *KDD*. 1984–1992.

[14] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NIPS*. 1024–1034.

[15] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. ACM.

[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[17] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*. 549–558.

[18] Chao Huang. 2021. Recent Advances in Heterogeneous Relation Learning for Recommendation. In *IJCAI*.

[19] 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.

[20] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. 2021. Knowledge-aware Coupled Graph Neural Network for Social Recommendation. In *AAAI*.

[21] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior recommendation with graph convolutional networks. In *SIGIR*. 659–668.

[22] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. IEEE, 197–206.

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

[24] Siwei Liu, Iadh Ounis, Craig Macdonald, and Zaiqiao Meng. 2020. A Heterogeneous Graph Neural Model for Cold-Start Recommendation. In *SIGIR*. 2029–2032.

[25] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *WWW*. ACM, 111–112.

[26] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*. 1441–1450.

[27] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *KDD*. 968–977.

[28] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *KDD*. 950–958.

[29] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. 165–174.

[30] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *WWW*. ACM, 2022–2032.

[31] Hongfa Wen, Xin Liu, Chenggang Yan, Linhua Jiang, Yaoqi Sun, Jiyong Zhang, and Haibing Yin. 2019. Leveraging Multiple Implicit Feedback for Personalized Recommendation with Neural Network. In *AIAM*. 1–6.

[32] Liang Wu, Diane Hu, Liangjie Hong, and Huan Liu. 2018. Turning clicks into purchases: Revenue optimization for product search in e-commerce. In *SIGIR*. 365–374.

[33] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *SIGIR*. 235–244.

[34] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *WSDM*. ACM, 153–162.

[35] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Bo Zhang, and Liefeng Bo. 2020. Multiplex Behavioral Relation Learning for Recommendation via Memory Augmented Transformer Network. In *SIGIR*. 2397–2406.

[36] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. 2021. Knowledge-Enhanced Hierarchical Graph Transformer Network for Multi-Behavior Recommendation. In *AAAI*.

[37] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *IJCAI*, Vol. 17. 3203–3209.

[38] Quanming Yao, Xiangning Chen, James T Kwok, Yong Li, and Cho-Jui Hsieh. 2020. Efficient neural interaction function search for collaborative filtering. In *WWW*. 1660–1670.

[39] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *KDD*. 793–803.

[40] Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. STAR-GCN: Stacked and Reconstructed Graph Convolutional Networks for Recommender Systems. In *IJCAI*.

[41] Yu Zheng, Chen Gao, Xiangnan He, Yong Li, and Depeng Jin. 2020. Price-aware Recommendation with Graph Convolutional Networks. In *ICDE*. IEEE, 133–144.

[42] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A neural autoregressive approach to collaborative filtering. In *ICML*.

[43] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, et al. 2020. A Knowledge-Aware Attentional Reasoning Network for Recommendation.. In *AAAI*. 6999–7006.