Coarse-to-Fine Knowledge-Enhanced Multi-Interest Learning Framework for Multi-Behavior Recommendation

CHANG MENG, Shenzhen International Graduate School, Tsinghua University, China
ZIQI ZHAO, School of Computer Science, Beihang University, China
WEI GUO, Huawei Singapore Research Center, Singapore
YINGXUE ZHANG, Huawei Technologies Canada, Canada
HAOLUN WU, McGill University, Canada
CHEN GAO, Department of Electronic Engineering, Tsinghua University, China
DONG LI, Huawei Noah's Ark Lab, China
XIU LI, Shenzhen International Graduate School, Tsinghua University, China
RUIMING TANG, Huawei Noah's Ark Lab, China

Multi-types of behaviors (e.g., clicking, carting, purchasing, etc.) widely exist in most real-world recommendation scenarios, which are beneficial to learn users' multi-faceted preferences. As dependencies are explicitly exhibited by the multiple types of behaviors, effectively modeling complex behavior dependencies is crucial for multi-behavior prediction. The state-of-the-art multi-behavior models learn behavior dependencies indistinguishably with all historical interactions as input. However, different behaviors may reflect different aspects of user preference, which means that some irrelevant interactions may play as noises to the target behavior to be predicted. To address the aforementioned limitations, we introduce multi-interest learning to the multi-behavior recommendation. More specifically, we propose a novel Coarse-to-fine Knowledge-enhanced Multi-interest Learning (CKML) framework to learn shared and behavior-specific interests for different behaviors. CKML introduces two advanced modules, namely *Coarse-grained Interest Extracting (CIE)* and *Fine-grained Behavioral Correlation (FBC)*, which work jointly to capture fine-grained behavioral dependencies. CIE uses knowledge-aware information to extract initial representations of each interest. FBC incorporates a dynamic routing scheme to further assign each behavior among interests. Empirical results on three real-world datasets verify the effectiveness and efficiency of our model in exploiting multi-behavior data.

CCS Concepts: • **Information systems → Recommender systems;**

Additional Key Words and Phrases: Multi-interest learning, multi-behavior recommendation, coarse-to-fine

C. Meng and Z. Zhao contributed equally to this research.

This work was partly supported by the Science and Technology Innovation 2030-Key Project (Grant No. 2021ZD0201404), Key Technology Projects in Shenzhen (Grant No. JSGG20220831110203007) and Aminer·ShenZhen·ScientificSuperBrain. And we thank MindSpore [1] for the partial support of this work, which is a new deep learning computing framework.

Authors' addresses: C. Meng and X. Li (corresponding author), Shenzhen International Graduate School, Tsinghua University, China; emails: mengc21@mails.tsinghua.edu.cn, li.xiu@sz.tsinghua.edu.cn; Z. Zhao, School of Computer Science, Beihang University, China; email: zziqi@buaa.edu.cn; W. Guo, Huawei Singapore Research Center, Singapore; email: guowei67@huawei.com; Y. Zhang, Huawei Technologies Canada, Canada; email: yingxue.zhang@huawei.com; H. Wu, McGill University, Canada; email: haolun.wu@mail.mcgill.ca; C. Gao, Department of Electronic Engineering, Tsinghua University, China; email: chgao96@gmail.com; D. Li and R. Tang (corresponding author), Huawei Noah's Ark Lab, China; emails: {lidong106, tangruiming}@huawei.com.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
© 2023 Copyright held by the owner/author(s).
1046-8188/2023/08-ART30 $15.00
https://doi.org/10.1145/3606369

ACM Reference format:
Chang Meng, Ziqi Zhao, Wei Guo, Yingxue Zhang, Haolun Wu, Chen Gao, Dong Li, Xiu Li, and Ruiming Tang. 2023. Coarse-to-Fine Knowledge-Enhanced Multi-Interest Learning Framework for Multi-Behavior Recommendation. *ACM Trans. Inf. Syst.* 42, 1, Article 30 (August 2023), 27 pages.
https://doi.org/10.1145/3606369

1 INTRODUCTION

Collaboration filtering (CF) [33] is widely used in industry to deeply probe into the latent information behind users' behaviors. CF firstly learns representations for both users and items from their historical interactions and then leverages these representations to make predictions. Most of the existing CF methods [17, 21, 22, 29, 39, 47] are designed to model a single behavior. However, users usually have interactions with items through various behaviors in real-world applications, like viewing, tagging as favorites, carting, and buying in the e-commerce scenario. As various behaviors may express users' complementary interests with items, utilizing the multi-behavior data simultaneously is necessary.

Many research efforts have been devoted to this problem to better capture collaborative signals from multi-behavior data, which can be divided into two categories [18]. The first category tackles the multi-behavior recommendation problem with advanced neural networks like attention network [14], transformer network [44, 45], and graph neural networks [11, 15, 37, 42, 45, 48]. The second category utilizes multi-behavior data with **multi-task learning (MTL)** [6]. These methods leverage all behaviors of users as prediction targets to improve the learning of users and items with different types of means, such as knowledge transfer [9, 12, 42] and graph neural networks [8].

However, these multi-behavior recommendation methods ignore the multi-faceted interests behind different behaviors. As shown in Figure 1(a), here we consider carting and buying, and exhibit a toy example. In this example, the user has a behavioral interaction with an item based on an interest (e.g., the user buys a hamburger based on his interest in "Junk Food"). Meanwhile, each behavior type is potentially attached to multiple interests (e.g., Buying is attached to "Junk Food" and "Electronic Goods"). Furthermore, we can observe that different interests that attached by the same behaviors (Cart) may have different effects on the prediction of target behaviors (Buy). Specifically, the interests "Electronic Goods" and "Luxury Goods" are both under the behavior of carting. However, for predicting whether the user will buy the computer, interest "Electronic Goods" is more effective and meaningful, because the user add the computer to cart based on "Electronic Goods" and add the less relevant item (watch) to cart based on "Luxury Goods". Here, we define interest "Electronic Goods" as shared interest for different behaviors as it is shared by carting and buying. Obviously, it is very meaningful to model the shared interest when correlating the interactive information of multiple behaviors. Besides, we define interest "Luxury Goods" which is specific only to carting as behavior-specific interest (same for interest "Junk Food"), and these specific interests may be noises for prediction of other behaviors. Fine-grained decoupling of behavior to interest-level representations can make full use of the potential dependence information in a delicate way, thus achieves both better interpretability and possibly superior performance. So it is vital to explore the relationships of multi-behaviors at a level of multi-interests.

Recent works have attempted to leverage multi-interest learning for recommendation. Some approaches implicitly cluster historical user interactions by using powerful encoders, like dynamic routing [7, 24, 41] and self-attention [7], while the others seek to leverage the auxiliary semantic information of knowledge graph to model multi-interests [5, 40]. Despite the effectiveness of these methods, they are all designed for single behavior, which share two common limitations when applied for multi-behavior recommendation:

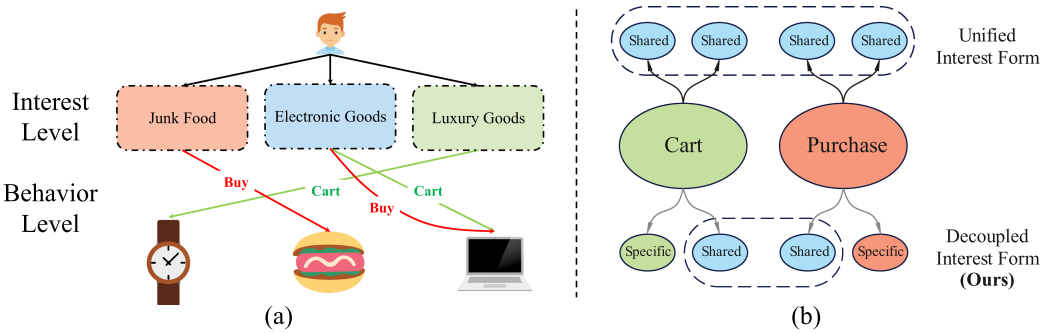

Fig. 1. (a) An example of multi-faceted interests behind multiple behaviors on an e-commerce scenario. Red and green represent cart and buy behaviors, respectively, and their specific interests are also represented in the same color. The blue represents the shared interests. (b) An example of behavior correlation at different granularities. Black and gray arrows represent different interest divisions, and dashed boxes represent correlations of multi-behavioral information at interest-level.

- **Inadequate Correlation Modeling.** Existing multi-interest methods are designed for single behavior, all of which adopts a group of unified interests for each user. However, if we unify all interests for each behavior to model behavior correlation, it may inevitably introduce noise, as some behavior-specific interests will negatively affect the learning of interest representations under other behaviors. We named this correlation modeling strategy as *unified interest form*, as shown in Figure 1(b). The *unified interest form* roughly correlate the divided shared interests of one behavior with the ones of other behaviors, which will lead to a inadequate modeling of correlation. So, it is necessary to design an interest extraction strategy that can fully consider the relationship between different behaviors.
- **Difficulties in Interest Learning.** The learning of interest can be regarded as the grouping of users' historical behaviors through *clustering*, where items from one cluster are expected to be closely related and collectively represent a specific aspect of user interests [24]. In clustering theory, the final results are sensitive to the *initialization of clustering centers*, which has been claimed in lots of works such as K-Means++ [3], K-Means|| [4], and Canopy [27]. Existing multi-interest methods like MIND [24] and DGCF [41] initialize clustering centers with random vectors, which lead to sub-optimal results as the generated centers may be very close to each other. To solve this problem, methods like KGIN [40] and KTUP [5] utilize the semantic information from knowledge graph to learn interest representation. However, they overlooks the rich collaborative signals that can be used for interest representation learning. As a result, a more flexible method which can make the initial centers of interest as far as possible by using semantic information obtained from knowledge-aware information and make full use of collaborative signals' information in clustering process is needed.

To tackle these two limitations, we propose a **C**oarse-to-fine **K**nowledge-enhanced **M**uti-interest **L**earning framework (**CKML**). To handle the inadequate modeling of behavioral correlations, CKML decouples interests into shared and behavior-specific parts for each behavior, then model the behavior correlation under the *decoupled interest form*, as shown in Figure 1(b).

To tackle the difficulties for interest learning process, CKML leverages a Coarse-to-fine strategy to initialize the interest centers and then allocates different interactions to different interests through the collaborative signals. Concretely, CKML consists of two modules: **C**oarse-grained **I**nterest **E**xtracting (**CIE**) module and **F**ine-grained **B**ehavioral **C**orrelation (**FBC**) module. For the first module, to capture the knowledge-aware item-item relations, it firstly learn representations

for interests under each behavior with the paradigm of graph neural networks, and then use the representations of knowledge-aware relations to initiate shared and behavior-specific interests for every behavior. Thus, this keeps the initial center of interest as far away as possible. For the second module, to adequately utilize the high-order user-item collaborative signals, we design a GNN-based framework with dynamic routing mechanism [30] to further finely allocate each interaction to different interests. In this module, we then generate fine-grained representations for all interests by graph propagation on separate interest-level graph. And finally, we only correlate the information of the shared interests of different behaviors through a self-attention mechanism to model the *decoupled interest form* correlation between different behaviors for multi-behavior prediction.

To summarize, this work makes the following contributions:

- We propose a novel CKML framework for multi-behavior recommendation, which learns shared and behavior-specific user interests for different behaviors. To the best of our knowledge, this is the first attempt to introduce multi-interest learning into the multi-behavior recommendation.
- We propose a multi-interest learning mechanism that models interest with a coarse-to-fine process. It contains a CIE module and a FBC module, which better models the complex dependencies among multiple behaviors.
- We conduct extensive experiments on three public datasets with a vastly different type of behaviors. The experimental results show the performance superiority and interpretability of our proposed framework.

2 RELATED WORK

In this section, we present the muti-behavioral methods appearing in recommendation based on the ways of modeling the representations of users and items, and introduce the existing multi-interest methods according to the different means of modeling interests.

2.1 Multi-Behavior Recommendation

The existing multi-behavior recommendation methods can be classified into two categories [18]. One is multi-behavioral representation modeling based on advanced neural networks, such as transformer and graph neural networks. For example, DIPN [14] proposes a hierarchical attention network, which uses both the intra-view and inter-view attention to learn the relationships between different behaviors. MATN [44] then uses transformer to encode the interactions of multiple behaviors, and proposes a memory augmented attention network which maps the context signals of different behaviors into representations of different spaces. While NMTR [12] proposes a new neural network model to extract multi-behavior data of users. Further, the GNN-based methods like MGNN [49], MBGCN [19], and GHCF [8] propose to leverage message passing on graphs to model high-order multi-behavioral interactive information. Besides, MBGMN [46] utilize meta network with GNN to capture high-order collaborative signals. Moreover, KHGT [45] combines GNN and transformer to model the global behavioral information, which not only captures the higher-order behavior between nodes but also addresses the dynamics of behavior. CML [42] combines meta contrastive learning and GNN to mine the higher-order information between nodes, effectively modeling individualized multi-behavior correlations.

The other category is to model different behaviors with MTL. DIPN [14], MGNN [49], and GHCF [8] regard the aggregated representations of different behaviors as shared input and use aggregated representations to predict each behavior individually. NMTR [12], EHCF [9], MBGMN [46], and CML [42] use a transfer learning paradigm to fully interact and aggregate different behavioral representations and then make predictions separately. All in all, these multi-behavior methods

try to capture the correlation between different behaviors, but they do not take into account the potential fine-grained interests behind each behavioral interaction. In contrast, our method makes full use of the behavioral correlation information of the interest level and alleviates the noise effect caused by coarse-grained modeling.

2.2 Multi-Interest Recommendation

Existing methods for multi-interest learning can be divided into two paradigms. One paradigm is utilizing collaborative behavioral signals to learn multi-interest representations. For example, MIND [24] applies a dynamic routing mechanism to assign each interaction to interests and uses label-aware attention to help learn user representations. On this basis, ComiRec [7] leverages self-attentive mechanism to extract user interests. To better learn interest representations, SINE [34] and Octopus [26] propose to model interests explicitly. They first build interest pools and then use attention mechanisms to explicitly activate some of the interests of users in the pool through historical user interactions. DGCF [41] introduces the dynamic routing mechanism into graphs with the modeling of independence among interests for multi-interest learning.

The other paradigm is leveraging structured relational information to construct multi-interest representations. For instance, KGIN [40] exploits the knowledge graph's structural information to learn the representations of different interests and aggregates information using GNN-based methods. KTUP [5] raises a new translation-based model, which leverages implicit interests to capture the relationship between users and items. To sum up, both paradigms have their own drawbacks. We can find that the former does not consider the importance of knowledge-aware information in the *initialization of interest clustering centers*, and the latter does not consider the importance of collaborative signals in the *process of interest clustering*. Our method not only well *initializes the interest clustering centers* but also sufficiently utilizes the collaborative signals to assist the *process of interest clustering*.

3 PROBLEM DEFINITION

3.1 List of Notations

The notations we used in this article are shown in Table 1.

3.2 Multi-Behavior Interaction Graph

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ represent the set of users and $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$ represent the set of items, where M and N are the numbers of users and items, respectively. In real-world recommendation scenarios, users can interact with items in multiple behaviors. Suppose there are K types of behaviors, we denote the user-item interaction data of different behaviors as $\mathcal{Y}_{u-i} = \{\mathbf{Y}^1_{u-i}, \mathbf{Y}^2_{u-i}, \ldots, \mathbf{Y}^K_{u-i}\}$, where $\mathbf{Y}^k_{u-i}$ represents the interaction matrix of behavior k, $y^k_{ui} = 1$ denotes that user u interacts with item i under behavior k, otherwise $y^k_{ui} = 0$. The user-item interaction data can also be regarded as a user-item bipartite graph $\mathcal{G}_{u-i} = (\mathcal{V}, \mathcal{E}_{u-i}, \mathcal{A}_{u-i}, \mathcal{R}_{u-i})$, where $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ is the node set containing all users and items, $\mathcal{E}_{u-i} = \cup_{k \in \mathcal{R}_{u-i}} \mathcal{E}^k_{u-i}$ is the edge set including all behavior records between users and items. Here k denotes a specific type of behavior and $\mathcal{R}_{u-i}$ is the set of all possible behavior types. $\mathcal{A}_{u-i} = \cup_{k \in \mathcal{R}_{u-i}} \mathbf{A}^k_{u-i}$ is the adjacency matrix set with $\mathbf{A}^k_{u-i}$ denoting adjacency matrix of a specific behavior graph $\mathcal{G}^k_{u-i} = (\mathcal{V}, \mathcal{E}^k_{u-i}, \mathbf{A}^k_{u-i})$.

3.3 Knowledge-Aware Relation Graph

To explore the rich semantics of items, we define graph $\mathcal{G}_{i-i} = (\mathcal{I}, \mathcal{E}_{i-i}, \mathcal{A}_{i-i}, \mathcal{R}_{i-i})$ to leverage side information like attributes and external knowledge to depict the multi-faced characteristics of items, the definition of $\mathcal{E}_{i-i}$ and $\mathcal{R}_{i-i}$ are similar to the definition of $\mathcal{E}_{u-i}$ and $\mathcal{R}_{u-i}$, respectively. We denote the item-item relation matrix as $\mathcal{A}_{i-i} = \{\mathbf{A}^1_{i-i}, \mathbf{A}^2_{i-i}, \ldots, \mathbf{A}^{|\mathcal{R}_{i-i}|}_{i-i}\}$, which can be constructed

Table 1. Notations and Corresponding Descriptions

Notation	Description
u, i	The user and item.
$\mathcal{U}, \mathcal{I}$	The set of users and items.
$\mathcal{V}$	The set of nodes on $\mathcal{G}_{u-i}$.
$\mathcal{G}_{u-i}, \mathcal{G}_{i-i}$	The user-item and item-item graphs.
$\mathcal{E}_{u-i}, \mathcal{E}_{i-i}$	The sets of edges on $\mathcal{G}_{u-i}$ and $\mathcal{G}_{i-i}$.
$\mathcal{A}_{u-i}, \mathcal{A}_{i-i}$	The set of adjacency matrices of $\mathcal{G}_{u-i}$ and $\mathcal{G}_{i-i}$.
$\mathcal{R}_{u-i}, \mathcal{R}_{i-i}$	The set of all possible behavior/relation types of $\mathcal{G}_{u-i}$ and $\mathcal{G}_{i-i}$.
$\mathbf{A}^k_{u-i}, \mathbf{A}^r_{i-i}$	The adjacency matrixes of behavior k and relation r.
$\mathbf{x}_u, \mathbf{y}_i$	The initialize embeddings for user u and item i.
$\mathbf{z}^r_i$	The learned representation of the ith item under the rth behavior in CIE.
$\mathbf{s}^k_i, \mathbf{h}^k_i$	The extracted behavioral-specific and shared interests embeddings of item i under behavior k in CIE.
$\mathbf{g}^k_i$	The output of CIE under behavior k. $\mathbf{g}^k_i = \mathbf{s}^k_i \| \mathbf{h}^k_i$.
N_{spe}, N_{sha}	The number of specific and shared interests for each behavior. Besides, $N_* = N_{spe} + N_{sha}$
d, d^*	The sizes of original embedding and interest embedding. Besides, $d^* = \frac{d}{N_{spe}} = \frac{d}{N_{sha}}$
$\mathbf{t}^k_{u-i}$	The time embedding for pair (u, i) under behavior k.
$\mathbf{a}^k_t$	The weight of edges on graph $\mathcal{G}^k_{u-i}$ at the tth iteration.
$\mathbf{e}^{k,l}_u, \mathbf{e}^{k,l}_i$	The input embedding at $(l+1)$-th layer (or the output embedding at lth layer) in FBC for user u and item i.
$\lambda^{u,h}_{k,k'}, \lambda^{i,h}_{k,k'}$	The relevance score between kth and $k'th$ behaviors of the hth head for the user u and item i.
$\mathbf{f}^k_{u,sha}, \mathbf{f}^k_{i,sha}$	The shared interests embeddings for user u and item i under behavior k before behavioral correlation.
$\mathbf{f}^{k,l}_{u,spe}, \tilde{\mathbf{f}}^{k,l}_{u,sha}$	The final specific and shared interests embeddings for user u under behavior k at the lth layer.
$\mathbf{f}^{k,l}_{i,spe}, \tilde{\mathbf{f}}^{k,l}_{i,sha}$	The final specific and shared interests embeddings for item i under behavior k at the lth layer.
$\hat{o}^k_{u,i}$	The prediction score for the pair (u, i) under behavior k.
$\hat{o}^r_{i,i'}$	The prediction score for the pair (i, i') under relation r.

with different reasons, such as items with same category, from the same restaurant, or interacted by similar users.

3.4 Task Description

Generally, there is a target behavior to be optimized (e.g., purchase), which we denote as $\mathbf{Y}^K_{u-i}$, and other behaviors $\{\mathbf{Y}^1_{u-i}, \mathbf{Y}^2_{u-i}, \ldots, \mathbf{Y}^{K-1}_{u-i}\}$ (e.g., view and tag as favorite) are treated as auxiliary behaviors for assisting the prediction of target behavior. The goal is to predict the probability that user u will interact with item i under target behavior K.

4 METHODOLOGY

We now present the model details of our proposed CKML, which is illustrated in Figure 2. It consists of two core modules: (1) CIE module, which utilizes knowledge-aware relations to extract shared and behavior-specific interests for multiple behaviors; and (2) FBC module, which allocates

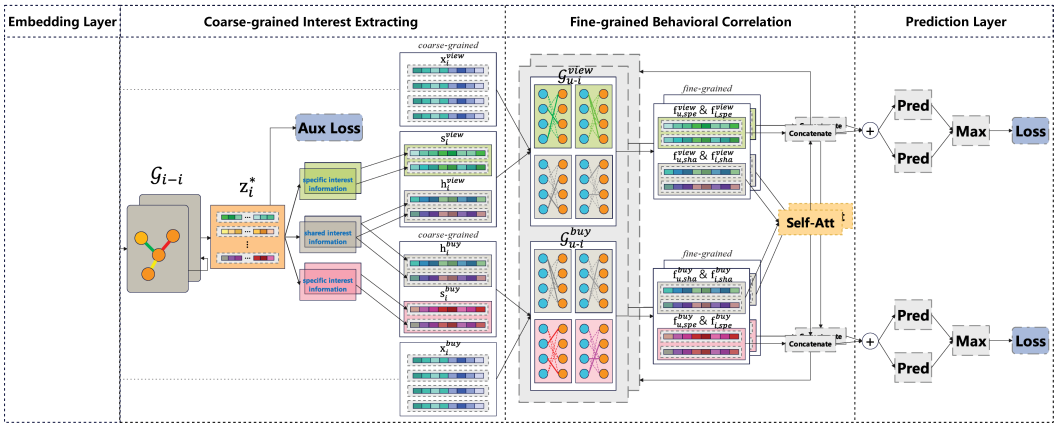

Fig. 2. Illustration of the proposed CKML. For brevity, only two behaviors (view and buy) are represented here. The green and red rectangles represent the behavior-specific interests, while the grey rectangle represents the shared interests. The orange circles represent items, while the blue circles represent users. (⊕) denotes the element-wise addition operation.

different interactions to different interests under each behavior, then models the complex behavior dependencies with interest-aware correlations.

4.1 Embedding Layer

In industrial applications, users and items are often denoted as high-dimensional one-hot vectors. Generally, given a user-item pair (u, i), we apply the embedding lookup operation for user u and item i to obtain the embedding vectors:

$$\mathbf{x}_u = \mathbf{E}_u^T \cdot \mathbf{p}_u, \ \mathbf{y}_i = \mathbf{E}_i^T \cdot \mathbf{p}_i, \tag{1}$$

where $\mathbf{E}_u \in \mathbb{R}^{M \times d}$ and $\mathbf{E}_i \in \mathbb{R}^{N \times d}$ are the created embedding tables for users and items, $\mathbf{p}_u \in \mathbb{R}^M$ and $\mathbf{p}_i \in \mathbb{R}^N$ denotes the one-hot IDs of user u and item i, and d is the embedding size.

4.2 Coarse-Grained Interest Extracting

Knowledge-aware item-item relations are widely used to supplement semantic information and assist representation learning [5, 40, 45]. Inspired by the strong semantics of relations in the knowledge-aware relation graph [38, 40, 45], we propose a CIE module to extract users' interests which motivates users' interactions of multiple behaviors. In this way, we obtain the initial interest clustering centers. To further verify that the initial centers of interest obtained by CIE are better than the randomly initialized centers of interest, we design experiments to visualize the output embeddings of CIE in Section 5.6.1. There are two main components in CIE: the first part is the knowledge-aware relation modeling which is designed for capturing the semantic information from the knowledge-aware item-item relation graph, while the second part is the behavior-aware interest extraction which is designed to utilize the semantic information obtained in the previous part to make an extraction of interests.

4.2.1 Knowledge-Aware Relation Modeling. Most existing multi-interest methods initialize interests with random generated vectors [24, 41], which fails to endow interests with semantics and may lead to a chaotic interest division. Since we have emphasized the importance of initializing interest clustering centers in Section 1, and inspired by the knowledge graph based methods [5, 40, 45], we use knowledge-aware information to initialize interest representations.

Thanks to its high capability in modeling relational data and great performance in representation learning, we seek to utilize knowledge-aware relations for interest extraction under the paradigm of graph neural networks in this component. Specially, we firstly partition the knowledge-aware relation graph $\mathcal{G}_{i-i}$ into several relation-specific sub-graphs $\mathcal{G}_{i-i}^1, \mathcal{G}_{i-i}^2, \ldots, \mathcal{G}_{i-i}^{|\mathcal{R}_{i-i}|}$, and the corresponding adjacency matrices are $\mathbf{A}_{i-i}^1, \mathbf{A}_{i-i}^2, \ldots, \mathbf{A}_{i-i}^{|\mathcal{R}_{i-i}|}$. As for the message propagation, we adopt the state-of-the-art GCN models, such as LightGCN [17], LR-GCCF [10], GCN [21], and NGCF [39], for graph information aggregation. And the neighbor propagation process in each layer of each sub-graph can be formulated as

$$\mathbf{z}_i^{r,l} = \underset{j \in N_i}{Agg} \left(\mathbf{z}_j^{r,l-1}, \mathbf{A}_{i-i}^r \right), \tag{2}$$

where r denotes the type of relation, l denotes the layer of GNN, N_i denotes the neighbors of item i, and $\mathbf{z}_i^{r,0} = \mathbf{y}_i$ is the initial embedding for item i. After the propagation, we average the generated representations from all layers to get the final representations:

$$\mathbf{z}_i^r = \sum_{l=0}^{L_{i-i}} \mathbf{z}_i^{r,l} / (L_{i-i} + 1), \tag{3}$$

where $\mathbf{z}_i^r \in \mathbb{R}^{1 \times d}$ and L_{i-i} is the number of GNN layers setting for modeling the knowledge-aware relation graph. We use the same number of layers for all relations here for simplicity.

4.2.2 Behavior-Aware Interest Extraction. Since we have obtained representations for all relations, how to effectively extract interests for different behaviors remains a challenge. As shown in Figure 1, different behaviors exhibit diverse interest patterns. Some interests are shared across multiple behaviors, while others are unique for specific behaviors. This is similar to shared expert information and specific expert information under different tasks in multi-task learning. Motivated by the customized gate presented in PLE [35], which achieves great performance in multi-task learning, we creatively propose to introduce shared interest and behavior-specific interest for multi-interest learning. The shared interest is designed to correlate with other types of behaviors at the level of interest, which can better leverage the potential complementary information of same interest within multiple behaviors. And the specific interest decouples and retains the independence of the corresponding behaviors, thus alleviating the influence of noise. We first combine the representations of all relations into a unified vector:

$$\mathbf{z}_i^* = \underset{r \in \mathcal{R}_{i-i}}{Concatenate}\, \mathbf{z}_i^r, \tag{4}$$

After that, we use a non-linear transformation which is generally used to model the combinations among relations to convert relations into multiple interests. For the specific interests, we have:

$$\mathbf{s}_i^k = \underset{s=1}{\overset{N_{spe}}{Concatenate}} \left(LeakyReLU \left(\mathbf{z}_i^* \cdot \mathbf{W}_s^k + \mathbf{b}_s^k \right) \right), \tag{5}$$

where N_{spe} is the number of specific interests for each behavior, s is the sth interest, $\mathbf{W}_s^k \in \mathbb{R}^{(|\mathcal{R}_{i-i}| * d) \times (\frac{d}{N_{spe}})}$ and $\mathbf{b}_s^k \in \mathbb{R}^{1 \times (\frac{d}{N_{spe}})}$ are transformation matrix and bias matrix, and $\mathbf{s}_i^k$ denotes the extracted behavioral-specific interests for behavior k. Notice that we use $\frac{1}{N_{spe}}$ of the original item embedding size as the interest size to keep similar space usage as single-interest models, and we apply the same compressed form to shared interests. For the behavioral shared interests, we have:

$$\mathbf{h}_i^k = \underset{s=1}{\overset{N_{sha}}{Concatenate}} \left(LeakyReLU \left(\mathbf{z}_i^* \cdot \mathbf{W}_s + \mathbf{b}_s \right) \right), \tag{6}$$

where N_{sha} is the number of shared interests, s is the sth interest, $\mathbf{W}_s \in \mathbb{R}^{(|\mathcal{R}_{i-i}|*d)\times(\frac{d}{N_{sha}})}$ and $\mathbf{b}_s \in \mathbb{R}^{1\times(\frac{d}{N_{sha}})}$ are transformation matrix and bias matrix, $\mathbf{h}_i^k$ denotes the extracted shared interests for behavior k. Since the parameters of different behaviors are shared, the shared representations of different k in this equation are consistent.

Finally, we union the representations of shared and specific interests as the output of CIE:

$$\mathbf{g}_i^k = \mathbf{s}_i^k \| \mathbf{h}_i^k, \tag{7}$$

where $(\|)$ is the concatenation operation between two vectors. For convenience, we set $\frac{d}{N_{spe}} = \frac{d}{N_{sha}} = d^*$, $N_* = N_{spe} + N_{sha}$.

4.3 Fine-Grained Behavioral Correlation

Existing multi-behavior methods model the dependencies among multiple behaviors without distinguishing the diverse interests on which different interactions are based, which may inevitably introduce noise if the interactions are due to different interests.

In the previous part, we have preliminarily extracted the potential interest of items based on the knowledge-aware relations. However, this is only a node-wise partitioning, and does not divide specific interactions (i.e., edges on the graph) into interests. Here "node-wise" means the level of users and items, while the corresponding "edge-wise" denotes a finer-grained level that considers each interaction between users and items. To address this problem, we propose a FBC layer to further allocate each interaction to different interests and model the dependence between behaviors at the level of interest. FBC is composed of two key components: The first one is interest-aware behavior allocation which is designed to further allocate each interaction to different interests. And the second one is interest-aware dependence modeling which is designed to capture inter-behavioral correlations and adequately leverage this information at each layer.

4.3.1 Interest-Aware Behavior Allocation. To allocate the edges on the graph $\mathcal{G}_{u-i}$ to different interests under each behavior, we apply the disentangled representation learning [7, 24, 41] for behavior allocation. We firstly partition the provided multi-behavior user-item graph $\mathcal{G}_{u-i}$ into behavior-specific sub-graphs $\mathcal{G}_{u-i}^1, \mathcal{G}_{u-i}^2, \ldots, \mathcal{G}_{u-i}^K$, and the corresponding adjacency matrices are $\mathbf{A}_{u-i}^1, \mathbf{A}_{u-i}^2, \ldots, \mathbf{A}_{u-i}^K$, which can be formulated as

$$\mathbf{A}_{u-i}^k = \begin{pmatrix} 0 & \mathbf{Y}_{u-i}^k \\ \left(\mathbf{Y}_{u-i}^k\right)^T & 0 \end{pmatrix}, \tag{8}$$

where $\mathbf{Y}_{u-i}^k$ is the user-item adjacency interaction matrix of behavior k, $\mathbf{A}_{u-i}^k \in \mathbb{R}^{(M+N)\times(M+N)}$, M and N denote the number of users and items, respectively. As for the processing of time, we simply follow KHGT [45], and first consider the edge $\mathcal{E}_{u-i}^k$ between user u and item i under behavior k, mapping their corresponding interaction timestamp t_{u-i}^k into the time slot as $\tau(t_{u-i}^k)$, then generate the embedding of time as $\mathbf{t}_{u-i}^k \in \mathbb{R}^{1\times d^*}$ for each interaction. Specifically, we have:

$$\begin{cases} \hat{\mathbf{t}}_{u-i}^{k,(2n)} = \sin\left(\dfrac{\tau(t_{u-i}^k)}{10000^{\frac{2n}{d}}}\right) \\ \hat{\mathbf{t}}_{u-i}^{k,(2n+1)} = \cos\left(\dfrac{\tau(t_{u-i}^k)}{10000^{\frac{2n+1}{d}}}\right) \\ \mathbf{t}_{u-i}^k = \hat{\mathbf{t}}_{u-i}^k \cdot \mathbf{W}_t \end{cases}, \tag{9}$$

Algorithm 1: The Allocation of Interests for the kth Behavior at Each Layer

Input : $\mathbf{A}_{u-i}^k \in \mathbb{R}^{(M+N)\times(M+N)}, \forall u \in \mathcal{U}, \forall i \in \mathcal{I}$:
$\mathbf{e}_u^k, \mathbf{e}_i^k \in \mathbb{R}^{N_*\times d^*}, \mathbf{t}_{u-i}^k \in \mathbb{R}^{1\times d^*}$

Output : Vector representations $\mathbf{f}_u^k$ for all $u \in \mathcal{U}$ and $\mathbf{f}_i^k$ for all $i \in \mathcal{I}$

/* Step 1. Initialize. */

$\mathcal{E}_{u-i}^k \leftarrow \mathbf{A}_{u-i}^k, \mathbf{a}_0^k \leftarrow \mathbf{1}^{N_*\times|\mathcal{E}_{u-i}^k|}$

$\mathbf{f}_{u,0}^k \leftarrow \mathbf{e}_u^k + \mathbf{1}^{N_*} \otimes \mathbf{t}_{u-i}^k, \mathbf{f}_{i,0}^k \leftarrow \mathbf{e}_i^k + \mathbf{1}^{N_*} \otimes \mathbf{t}_{u-i}^k$

for $t = 1$ *to* N_{iter} **do**

  /* Step 2. Normalize coefficients. */

  $\mathbf{a}_t^k \leftarrow Softmax(\mathbf{a}_{t-1}^k/\tau)$

  for $s = 1$ *to* N_* **do**

    /* Step 3. Embedding propagation. */

    $\mathbf{f}_{u,t}^k[s] \leftarrow \sum_{\mathcal{E}_{u-i}^k} \mathbf{a}_t^k[s] \cdot \mathbf{f}_{i,0}^k[s], \forall u \in \mathcal{U}$

    $\mathbf{f}_{i,t}^k[s] \leftarrow \sum_{\mathcal{E}_{u-i}^k} \mathbf{a}_t^k[s] \cdot \mathbf{f}_{u,0}^k[s], \forall i \in \mathcal{I}$

    /* Step 4. Update coefficients. */

    $\Delta\mathbf{a}_t^k[s] \leftarrow \underset{(p,q)\in\mathcal{E}_{u-i}^k}{Stack} \left(\frac{\mathbf{f}_{q,0}^k[s]}{|\mathbf{f}_{q,0}^k[s]|} \cdot tanh\left(\frac{\mathbf{f}_{p,t}^k[s]}{|\mathbf{f}_{p,t}^k[s]|}\right)^\top\right)$

  end

end

/* Step 5. Information aggregation. */

$\mathbf{f}_u^k \leftarrow Agg(\underset{s=1}{\overset{N_*}{Concatenate}}(\mathbf{f}_{u,N_{iter}}^k[s]), \mathbf{A}_{u-i}^k), \forall u \in \mathcal{U}$

$\mathbf{f}_i^k \leftarrow Agg(\underset{s=1}{\overset{N_*}{Concatenate}}(\mathbf{f}_{i,N_{iter}}^k[s]), \mathbf{A}_{u-i}^k), \forall i \in \mathcal{I}$

where the element indexs (even and odd position index) in the temporal information embedding are represented as $2n$ and $2n+1$, respectively. $\mathbf{W}_t \in \mathbb{R}^{2d\times d^*}$ is the transformation weights corresponding to k-th type of interactions.

To better illustrate the process of the allocation of interests on each layer, we take the kth behavior as an example. As shown in Algorithm 1, we use $\mathcal{E}_{u-i}^k = \{(p,q)|\mathbf{A}_{u-i}^k[p,q] \neq 0\}$ to represent the set of edges on graph $\mathcal{G}_{u-i}^k$. Meanwhile, we set $\mathbf{a}_0$ as the initial weight for each edge on $\mathcal{G}_{u-i}^k$ and initialize the embedding for each user and item. We leverage a kronecker product ($\otimes$) to replicate the vector $\mathbf{t}_{u-i}^k$ for N_* times along the row direction and add it to $\mathbf{e}_u^k$ and $\mathbf{e}_i^k$, thus get $\mathbf{f}_{u,0}^k$ and $\mathbf{f}_{i,0}^k$ (Step 1). Here, for simplicity, we denote $\mathbf{e}_u^k$ and $\mathbf{e}_i^k$ as the output of the previous layer. Next, we start iterative process. In the tth iteration, in order to get distributions across all interests, we use the softmax function to normalize these coefficients (Step 2):

$$\mathbf{a}_t^k[s] = \frac{\exp\left(\mathbf{a}_{t-1}^k[s])/\tau\right)}{\sum_{s=1}^{N_*} \exp\left(\mathbf{a}_{t-1}^k[s]/\tau\right)}, \tag{10}$$

where $\mathbf{a}_t^k$ denotes the vector of weight coefficients of each edge of graph $\mathcal{G}_{u-i}^k$ in the tth iteration, τ is the temperature coefficient, s denotes the sth interest. Furthermore, in each iteration, we assign all the edges on the graph $\mathcal{G}_{u-i}^k$ to each interest of users and items on the graph (Step 3). At this step, $\mathbf{f}_{u,t}^k[s]$ and $\mathbf{f}_{i,t}^k[s]$ represent the sth interest for user u and item i after the allocation of the edge weights, respectively. Last but not least, we calculate the affinity between each pairs of nodes on the graph $\mathcal{G}_{u-i}^k$ to update the weight of each edge (Step 4). Here, $\mathbf{a}_t^k[s]$ denotes the updated

weight of edges at the tth iteration of the sth interest under behavior k. After all of the iterations, we finally take the representation generated by the last iteration as the final output, and aggregate them with GCN models (Step 5), which is the same as the aggregators in Section 4.2.1.

4.3.2 Interest-Aware Behavioral Correlation. After the allocation of interests for every node at each layer, we have got $\mathbf{f}_u^k = \mathbf{f}_{u,spe}^k \| \mathbf{f}_{u,sha}^k$ and $\mathbf{f}_i^k = \mathbf{f}_{i,spe}^k \| \mathbf{f}_{i,sha}^k$. Furthermore, we need to correlate information between behaviors at the interest level. And we just correlate the information between the representations of *shared* interests of each behavior with a self-attention network [36] because the behavior-specific interests contain few useful information for the target behavior and may contain noise. For instance, in the Yelp dataset, there are behaviors (Dislike) that are contrary to the target behavior (Like), which may interfere with the learning of target behavior. For better convergence, we apply residual connection to the output of self-attention [16], which can be formulated as

$$\begin{cases} \tilde{\mathbf{f}}_{u,sha}^k = \mathbf{MH} - \text{Att}\left(\mathbf{f}_{u,sha}^k\right) + \sum_{k'=1}^{K} \mathbf{f}_{u,sha}^{k'} \\ \mathbf{MH} - \text{Att}\left(\mathbf{f}_{u,sha}^k\right) = \underset{h=1}{\overset{H}{Concatenate}}\left(\sum_{k'=1}^{K} \lambda_{k,k'}^{u,h} \cdot \tilde{\mathbf{V}}^h \cdot \mathbf{f}_{u,sha}^{k'}\right) \\ \lambda_{k,k'}^{u,h} = Softmax(\bar{\lambda}_{k,k'}^{u,h}) \\ \bar{\lambda}_{k,k'}^{u,h} = \dfrac{\left(\tilde{\mathbf{Q}}^h \cdot \mathbf{f}_{u,sha}^k\right)^\top \left(\tilde{\mathbf{K}}^h \cdot \mathbf{f}_{u,sha}^{k'}\right)}{\sqrt{d^*/H}} \end{cases}, \tag{11}$$

where $\tilde{\mathbf{Q}}^h, \tilde{\mathbf{K}}^h, \tilde{\mathbf{V}}^h \in \mathbb{R}^{\frac{d^*}{H} \times \frac{d^*}{H}}$ are learnable projection matrices of the h-th head. $\lambda_{k,k'}^{u,h}$ represents the relevance score between kth and $k'th$ behaviors of the hth head for the user u. Moreover, similar operations are applied for the item i.

Finally, for the information propagation of the kth behavior, we have:

$$\begin{cases} \mathbf{e}_u^{k,l} = \mathbf{f}_{u,spe}^{k,l} \| \tilde{\mathbf{f}}_{u,sha}^{k,l} + \mathbf{e}_u^{k,l-1}, \forall u \in \mathcal{U} \\ \mathbf{e}_i^{k,l} = \mathbf{f}_{i,spe}^{k,l} \| \tilde{\mathbf{f}}_{i,sha}^{k,l} + \mathbf{e}_i^{k,l-1}, \forall i \in \mathcal{I} \end{cases}, \tag{12}$$

where $l \in [1, \ldots, L_{u-i}]$, L_{u-i} is the number of GNN layer, $(\|)$ is the concatenated operation for two vectors. $\mathbf{e}_u^{k,0} = \mathbf{x}_u^k = \mathbf{x}_u$ and $\mathbf{e}_i^{k,0} = \mathbf{g}_i^k$.

4.4 Joint Optimization

4.4.1 The Prediction of the U-I Interaction. In the above parts, we have obtained the shared and behavior-specific representations $\mathbf{f}_{u,spe}^{k,l}$ and $\tilde{\mathbf{f}}_{u,sha}^{k,l}, \forall l \in [1, 2, \ldots, L_{u-i}], \forall k \in [1, 2, \ldots, K], \forall u \in \mathcal{U}$, similar operations are applied for the item i. To aggregate the information of each layer, we follow KHGT [45], and simply add them up. Thus we have:

$$\begin{cases} \mathbf{f}_u^{k,*} = \sum_{l=1}^{L_{u-i}} (\mathbf{f}_{u,spe}^{k,l} \| \tilde{\mathbf{f}}_{u,sha}^{k,l}), \forall u \in \mathcal{U} \\ \mathbf{f}_i^{k,*} = \sum_{l=1}^{L_{u-i}} (\mathbf{f}_{i,spe}^{k,l} \| \tilde{\mathbf{f}}_{i,sha}^{k,l}), \forall i \in \mathcal{I} \end{cases}, \tag{13}$$

where $\mathbf{f}_u^{k,*}, \mathbf{f}_i^{k,*} \in \mathbb{R}^{N_* \times d^*}$, k represents the kth behavior. Inspired by ComiRec [7], we make separate predictions for each interest under each behavior and take the maximum of all the predictions

under each behavior, which can be formulated as

$$\hat{o}_{u,i}^{k} = \max_{s=1}^{N_*}\left(\sum_{j}^{d^*}(\mathbf{f}_u^{k,*}[s] \circ \mathbf{f}_i^{k,*}[s])[j]\right), \tag{14}$$

where $s \in [1, 2, \ldots, N_*]$ denotes the sth interest, $(\circ)$ is the hadamard product operation.

Finally, to perform the model optimization, we follow KHGT [45] and use marginal pair-wise ***Bayesian Personalized Ranking*** (**BPR**) loss function to minimize the following loss function:

$$\mathcal{L}_{u-i} = \sum_{k=1}^{K} \sum_{(u,p,q)\in O_{u-i,k}} \alpha^k * \max\left(0, 1 - \hat{o}_{u,p}^{k} + \hat{o}_{u,q}^{k}\right), \tag{15}$$

where the $\alpha^k \in [0, 1]$ denotes the coefficient of loss for kth behavior, $O_{u-i,k} = \{(u,p,q)|(u,p) \in O_{u-i,k}^{+}, (u,q) \in O_{u-i,k}^{-}\}$ denotes the training dataset. $O_{u-i,k}^{+}$ indicates observed positive user-item interactions under behavior k and $O_{u-i,k}^{-}$ indicates unobserved user-item interactions under behavior k.

4.4.2 The Prediction of the Knowledge-Aware Item-Item Relation. Inspired by self-supervised learning on graphs [43], we use the information of item-item relations to reconstruct the item-item graphs, which can be considered as a **self-supervised relation reconstruction** (**SRR**) task to enhance the learning of interest representations.

In detail, since we obtained the representations of each relation for all item $i \in \mathcal{I}$, i.e., $\mathbf{z}_i^*$ at Section 4.2.1, we calculate prediction scores for each relation between items:

$$\hat{o}_{i,i'}^{r} = \sum_{j}^{d}(\mathbf{z}_i^r \circ \mathbf{z}_{i'}^r)[j], \tag{16}$$

where r represents the rth relation. We then use the BPR loss to reconstruct the graph $\mathcal{G}_{i-i}^r$, which can be formulated as

$$\mathcal{L}_{i-i} = -\sum_{r=1}^{|\mathcal{R}_{i-i}|} \sum_{(i,p,q)\in O_{i-i,r}} \ln \sigma\left(\hat{o}_{i,p}^{r} - \hat{o}_{i,q}^{r}\right), \tag{17}$$

where $O_{i-i,r} = \{(i,p,q)|(i,p) \in O_{i-i,r}^{+}, (i,q) \in O_{i-i,r}^{-}\}$ denotes the training dataset of the item-item relation graph reconstructive task, which is similar to the definition in Section 4.4.1. Finally, for the total loss, we have:

$$\mathcal{L}_{total} = \mathcal{L}_{u-i} + \beta\mathcal{L}_{i-i} + \lambda\|\Theta\|_{\mathrm{F}}^2, \tag{18}$$

where Θ represents the set of all trainable parameters, λ is the weight for the regularization term, $\beta \in [0, 1]$ is the weight of $\mathcal{L}_{i-i}$.

4.5 Complexity Analysis

4.5.1 Time Complexity. In CIE, we spend $O(L_{i-i}|\mathcal{E}_{i-i}|d)$ for message propagation in the knowledge-aware item-item graph, where L_{i-i} denotes the number of GNN layers in handling item-item relations, $|\mathcal{E}_{i-i}|$ is the number of edges on $\mathcal{G}_{i-i}$ and d is the embedding size. After that, the time spent to extract the interest from the item-item relation is $O(|\mathcal{R}_{i-i}|d^2)$, where $|\mathcal{R}_{i-i}|$ refers to the number of relations. In FBC, it takes $O(L_{u-i}|\mathcal{E}_{u-i}|d)$ to propagate embedding in the user-item bipartite graph, where L_{u-i} is the number of GNN layers in handling user-item relations and $|\mathcal{E}_{u-i}|$ denotes the number of edges on $\mathcal{G}_{u-i}$. Besides, the computational complexity of the self-attention mechanism is $O(KL_{u-i}d^2)$, where K is the number of behaviors. In summary, the overall time complexity of CKML mainly comes from the GNN part. The time complexity of our

model is comparable to other GNN-based methods and we perform experiments to validate it in Section 5.2.3.

4.5.2 Space Complexity. Most of the parameters that the model needs to learn are the embedding of the user and item, which costs $O((M+N)*d)$. The space size of the transformation matrixs in extracting shared interests and specific interests are $O(|\mathcal{R}_{i-i}|d^2 + d)$ and $O(K|\mathcal{R}_{i-i}|d^2 + Kd)$, where K is the number of behaviors. The space size of $\tilde{\mathbf{Q}}^h$, $\tilde{\mathbf{K}}^h$ and $\tilde{\mathbf{V}}^h$ in the attentional mechanism requires $O(L_{u-i} * \frac{(d^*)^2}{H})$, where the $d^* = \frac{d}{N_{spe}} = \frac{d}{N_{sha}}$, H is the number of attention head. All in all, CKML have limited additional parameters except for the embedding of the user and item.

5 EXPERIMENTS

We conduct experiments to answer the following questions:

- **RQ1:** How does CKML perform in terms of effectiveness and efficiency against various baselines?
- **RQ2:** How do different components of CKML affect the performance?
- **RQ3:** Can the design of shared and behavior-specific interests bring benefits to multi-behavior recommendation?
- **RQ4:** How do different hyper-parameters affect the performance of CKML?
- **RQ5:** How is the interest interpretability of CKML? Are the cluster centers of different interests really farther apart? Can the shared and specific interest patterns captured by CKML be represented in an explainable way?

5.1 Experimental Setting

5.1.1 Dataset Description. We evaluate our model on three public datasets (i.e., **Yelp**,[1] **Online Retail**[2] and **Tmall**[3]) with the same parameter settings and preprocessing as compared baseline models. The behavior types and statistics of the three datasets are shown in Table 2.

- **Yelp:** We have fully aligned the experiment protocol with KHGT. And following the partition strategy in References [25, 28], KHGT differentiates the explicit user-item interactive behavior into three types in terms of user rating scores (i.e., ranging from 1 (worst) to 5 (best) stars with 0.5 star as increment), i.e., *dislike* ($r_{scores} \in [0, 2]$), *neutral* ($r_{scores} \in (2, 4)$), and *like* ($r_{scores} \in [4, 5]$). In addition, users offer tips about venues, which are considered as *tips* behaviors.
- **Online Retail:** Following KHGT, we also tested our CKML on a real-world online retail dataset containing explicit user-item interactions of multiple types, which includes *page view*, *add-to-cart*, *favorite*, and *purchase*.
- **Tmall:** The dataset is collected from Tmall, which is one of China's largest e-commerce platforms. It contains various user interactions, including *page views*, adding items to *favorites* or *carts*, and making *purchases*. To follow the approach taken in CML [42], we only included users with at least three purchases for our training and testing datasets.

Following the setting of KHGT and CML, *like* is regarded as the target behavior, i.e., the behavior to be predicted, for Yelp, while *purchase* is the target behavior of Online Retail and Tmall.

5.1.2 Evaluation Protocols. We apply two widely used metrics i.e., Hit Ratio (HR@N) and Normalized Discounted Cumulative Gain (NDCG@N) to evaluate the performance. *HR@N* is a

[1]https://www.yelp.com/dataset/download.
[2]https://github.com/akaxlh/KHGT.
[3]https://github.com/weiwei1206/CML.

Table 2. Statistics of Evaluation Datasets

Dataset	#User	#Item	#Interaction	#Target Interaction	#Interactive Behavior Type
Yelp	19,800	22,734	1.4×10^6	677,343	{Tip, Dislike, Neutral, Like}
Online Retail	147,894	99,037	7.7×10^6	642,916	{Page View, Favorite, Cart, Purchase}
Tmall	31,882	31,232	1.5×10^6	167,862	{Page View, Favorite, Cart, Purchase}

recall-based metric which measures the average proportion of right items in the top-k recommendation lists. On the other hand, *NDCG@N* evaluates the ranking quality of the top-k recommendation lists in a position-wise manner. To fairly compare our models and baselines, we follow the evaluation settings of KHGT, and set $N = 10$ by default in all experiments. Following the setting of KHGT, the last interacted item on the behavior to be predicted is used as a positive example in the test data, while the 99 randomly selected items the user has not interacted with are taken as negative examples. Besides, we also provide an all-item ranking [23, 31] to evaluate the performance of the recent recommender algorithms.

5.1.3 Baseline Models. To verify the effectiveness of our CKML model, we compare it with various baseline models, which can be categorized into four groups: (A) Single-behavior non-graph models (BPR [29], AutoRec[4] [32], MIND [24], ComiRec[5] [7]); (B) Single-behavior graph models (NGCF[6] [39], DGCF [41], KGAT[7] [38]); (C) Multi-behavior non-graph models (NMTR [12], DIPN [14], MATN[8] [44]); (D) Multi-behavior graph models (DGCF_M [41], NGCF_M [39], LightGCN_M[9] [17], MBGCN[10] [19], CML[11] [42], KHGT[12] [45]). Among them, MIND, ComiRec, and DGCF are multi-interest based models, MATN and KHGT are transformer-based models, and CML is contrastive learning based model. As DGCF, NGCF, and LightGCN are originally designed for single behavior, we use the multi-behavior as input to train these models and name them DGCF_M, NGCF_M, and LightGCN_M.

Single-behavior Non-graph Models:

- **BPR** [29] It is a conventional approach to collaborative filtering that utilizes pairwise ranking loss to personalize item recommendations and generate item rankings.
- **AutoRec** [32] It encodes vectors of users and items through reconstruction functions based on the autoencoder framework.
- **MIND** [24] It designs a multi-interest extractor layer with a variant dynamic routing to extract users' diverse interests and uses a label-aware attention scheme to learn these interests.
- **ComiRec** [7] It captures multiple interests from interactions of users, retrieving candidate items from the large-scale item pool. Besides, this method leverages a controllable factor to balance the recommendation accuracy and diversity.

Single-behavior Graph Models:

- **NGCF** [39] This approach exploits higher-order connectivity of user-item bipartite graphs via GNN.

[4]https://github.com/gtshs2/Autorec.
[5]https://github.com/THUDM/ComiRec.
[6]https://github.com/xiangwang1223/neural_graph_collaborative_filtering.
[7]https://github.com/xiangwang1223/knowledge_graph_attention_network.
[8]https://github.com/akaxlh/MATN.
[9]https://github.com/kuandeng/LightGCN.
[10]https://github.com/tsinghua-fib-lab/MBGCN.
[11]https://github.com/weiwei1206/CML.
[12]https://github.com/akaxlh/KHGT.

— **DGCF** [41] This model disentangles user-item interaction diagrams by modeling the interests behind the interactions, aiming to learn the representations of different interests.
— **KGAT** [38] It uses the GAT framework, captures higher-order connectivity between user and item in a collaborative knowledge graph, which combines user-item interaction graphs and knowledge graphs.

Multi-behavior Non-graph Models:

— **NMTR** [12] It captures cascading relationships between users' multi-behavioral interactions using multi-task learning.
— **DIPN** [14] This method leverages multiple behavioral interactions to predict user purchase intention via recurrent neural network and attention mechanism.
— **MATN** [44] It explores the dependencies between multiple behaviors and the contribution on target behavior.

Multi-behavior Graph Models:

— **DGCF$_M$** [41] It takes multi-behavioral interactive information as input, and correlates the information of different behaviors at interest-level through attention mechanism.
— **NGCF$_M$** [39] It utilizes multiple behaviors by modeling the relationship between multiple behaviors according to KHGT.
— **LightGCN$_M$** [17] It removes feature transformation and nonlinear activation from GCN. Each category of behavior has the same influence on the target behavior.
— **MBGCN** [19] This method uses graph convolutional networks on multi-behavior user-item interaction graphs, which learn the weights of multiple behaviors during embedding propagation.
— **CML** [42] This approach proposes meta-learning and contrastive meta-learning paradigms to distill transferable knowledge across different types of behaviors.
— **KHGT** [45] It encodes multi-behavioral interactive information between user and item using Graph Transformer Network to and infer the influence of multi-behavior interactions on target behavior.

5.1.4 Parameter Settings. Our proposed CKML is implemented in TensorFlow [2]. We fix the embedding size as 16 in line with KHGT for a fair comparison. The batch size is searched in {16,32,64}. We initialize the parameters using Xavier [13]. The parameters are optimized by Adam [20], while the learning rate and decay rate are set to 10^{-3} and 0.96, respectively. We search the number of GNN layers in {1,2,3,4} for the knowledge-aware item-item graph and user-item bipartite graph, respectively. We set the number of self-attention head to 2. The number of shared interests is varied in {1,2,4} as well as the number of specific interests, which is investigated in Section 5.5.1. The temperature coefficient used in the interest-aware behavior allocation is tuned in {0.1,1,5,10,20}, and the corresponding number of iterations is set to 2. We conduct a grid search of the loss coefficient for each behavior in {0,0.2,0.4,0.6,0.8,1}. All experiments are run for 5 times and average results are reported.

5.2 Performance Comparison (RQ1)

5.2.1 Effectiveness Comparison under the Setting of 99 Negative Samples. Table 3 shows the performance of different methods on three datasets with respect to HR@10 and NDCG@10. We have the following findings:

The effectiveness of CKML model. Our proposed CKML consistently achieves the best results on all datasets. More specifically, CKML improves the strongest baselines by **1.82%**, **5.61%** and

Table 3. The Overall Performance Comparison for Sampling-Item Test

Model	Yelp		Retail		Tmall	
	HR	NDCG	HR	NDCG	HR	NDCG
BPR	0.744	0.450	0.261	0.165	0.244	0.150
AutoRec	0.765	0.472	0.313	0.190	0.321	0.156
MIND	0.789	0.514	0.307	0.191	0.314	0.185
ComiRec	0.774	0.488	0.314	0.196	0.291	0.184
NGCF	0.789	0.500	0.302	0.185	0.314	0.173
DGCF	0.861	0.587	0.304	0.169	0.322	0.184
KGAT	0.835	0.543	0.377	0.214	0.395	0.243
NMTR	0.790	0.478	0.332	0.179	0.362	0.215
DIPN	0.791	0.500	0.317	0.178	0.323	0.207
MATN	0.826	0.530	0.354	0.209	0.406	0.225
DGCF_M	0.863	0.591	0.467	0.282	0.448	0.280
NGCF_M	0.793	0.492	0.374	0.221	0.322	0.182
LightGCN_M	0.873	0.573	0.472	0.277	0.455	0.282
MBGCN	0.796	0.502	0.369	0.222	0.381	0.213
CML	0.785	0.471	0.499	0.289	0.513	0.302
KHGT	0.880	0.603	0.464	0.278	0.391	0.232
CKML	**0.896***	**0.624***	**0.527***	**0.323***	**0.527***	**0.321***
Rel Impr.	1.82%	3.48%	5.61%	11.76%	2.73%	6.29%

Boldface denotes the highest score and underline indicates the results of the best baselines.
$\star$ represents significance level p-value < 0.05 of comparing CKML with the best baseline.

2.73% in terms of HR (**3.48%**, **11.76%**, and **6.29%** in terms of NDCG) on Yelp, Retail, and Tmall datasets, respectively. The great improvements over baselines demonstrate the effectiveness of CKML for multi-behavior recommendation.

Both GNN and multi-behavior based methods improve model performance. Despite the various architectures among different baseline models, we can find that GNN based models have a consistent trend that perform much better than non-graph models. For example, by incorporating neighbor information into representations, MBGCN and NGCF outperform DIPN and BPR in most datasets and metrics at the multi-behavior and single-behavior settings, respectively. Besides, multi-behavior models KHGT and MBGCN achieve much better performance than single-behavior model KGAT and NGCF, which further verifies the effectiveness of adding multi-behavior information for learning.

CKML consistently outperforms GNN based multi-behavior baseline models. Our proposed CKML surpasses the performance of DGCF_M, NGCF_M, LightGCN_M, MBGCN, and the state-of-the-art multi-behavior model KHGT and CML. By empowering the multi-behavior recommendation with multi-interest learning, CKML is capable of modeling the complex dependencies among multiple behaviors with multi-grained representations to infer user preference. While existing multi-behavior models only consider the observed user-item interactions as unified representations. Noticed that CML performs well on Retail and Tmall datasets but significantly worse on the other two datasets. A probable reason is that behaviors in Yelp are mutually exclusive (e.g., Dislike and Like), while CML assumes that different behaviors of the same user are similar for contrastive learning, which is unreasonable.

5.2.2 Effectiveness Comparison under the Setting of All-Item Ranking. All-item ranking is another evaluation protocol which is widely used for testing [23, 31]. For comprehensive comparison,

Table 4. The Overall Performance Comparison for All-Item Test

Model	Yelp		Retail		Tmall	
	HR	NDCG	HR	NDCG	HR	NDCG
MIND	0.0171	0.0087	0.0074	0.0037	0.0093	0.0047
ComiRec	0.0320	0.0156	0.0073	0.0039	0.0090	0.0042
NGCF	0.0230	0.0108	0.0033	0.0018	0.0086	0.0043
NGCF_M	0.0317	0.0146	0.0061	0.0029	0.0100	0.0048
CML	0.0320	0.0150	0.0103	0.0049	0.0140	0.0063
KHGT	0.0428	0.0213	0.0099	0.0051	0.0102	0.0053
CKML	**0.0558***	**0.0265***	**0.0122***	**0.0061***	**0.0177***	**0.0082***
Rel Impr.	30.37%	24.41%	18.45%	19.61%	26.43%	30.16%

Boldface denotes the highest score and underline indicates the results of the best baselines.
$\star$ represents significance level p-value < 0.05 of comparing CKML with the best baseline.

Table 5. Training Time Comparison (Seconds Per Epoch) of Different Methods on All Three Datasets

Training time (s) / Dataset / Model	Yelp	Retail	Tmall
KHGT	46.65	74.31	61.05
CKML	40.29	59.88	48.77

we compare our CKML with advanced methods under this setting. Specifically, we take the last item in the test data that interacts with the behavior to be predicted as a positive example, and all of the items that users do not interact with as the negative examples. As shown in Table 4, we can find that our CKML still performs best under this setting. Specifically, CKML improves the strongest baselines by **30.37%**, **18.45%**, and **26.43%** in terms of HR (**24.41%**, **19.61%**, and **30.16%** in terms of NDCG) on Yelp, Retail, and Tmall datasets, respectively. The results show that our model has good robustness under different ranking settings.

5.2.3 Efficiency Comparison. In addition to effectiveness, efficiency is also important. We conduct experiments to evaluate the cost of time of training and testing. Each result is obtained while the models are training in a single cluster, where each node contains 16 cores Intel(R) Xeon(R) Silver 4216 CPU (2.10 GHz) as well as 1 NVIDIA GeForce RTX 3090. And the following are details.

- **Training Efficiency.** Table 5 shows the average training time of our proposed CKML and KHGT for each epoch. For the sake of fairness, we set the parameters related to training efficiency consistent, like batch size and GNN layer. We can find that CKML is faster with **13.63%**, **19.42%**, and **20.11%** time reduction on the three datasets. One probable reason is that we split the complete graph into several smaller graphs under interests, and then make computation separately on these smaller graphs, which can be accelerated by parallel computation.
- **Testing Efficiency.** For the sake of fairness, we set the parameters related to testing efficiency consistent, like batch size and GNN layer. As shown in Table 6, we can find that our proposed CKML is **12.59%**, **9.42%**, and **21.00%** faster than KHGT on the three datasets for the testing time. The results show that our proposed CKML has higher efficiency when tested on the three datasets, which further demonstrates our views.

Table 6. Testing Time Comparison (Seconds Per Epoch) of Different Methods on All Three Datasets

Testing time (s) / Model \ Dataset	Yelp	Retail	Tmall
KHGT	41.05	171.45	91.05
CKML	35.88	155.30	71.93

Table 7. Performance of Different CKML Variants

Model	Yelp		Retail		Tmall	
	HR	NDCG	HR	NDCG	HR	NDCG
CKML	**0.896***	**0.624***	**0.527***	**0.323***	**0.527***	**0.321***
CKML w/o CIE	0.893	0.619	0.510	0.310	0.507	0.308
CKML w/o FBC	0.887	0.610	0.491	0.290	0.508	0.311
CKML w/o MI	0.839	0.524	0.444	0.246	0.387	0.227

$\star$ represents significance level p-value < 0.05 of comparing CKML with other variants.
Boldface denotes the highest score.

In summary, we claim that CKML has the best overall training and testing efficiency.

5.3 Ablation Study (RQ2)

CKML is built with several important designations including the Multi-Interest (MI), the CIE and the FBC. To analyze the rationality of each design consideration, we explore CKML with several different model variants.

- **CKML w/o CIE:** We remove the coarse-grained interest extracting module and express each interest with randomly initialized vectors.
- **CKML w/o FBC:** We replace the fine-grained behavioral correlation module with a combination of the best-performing GCN methods (LigthGCN for Yelp, GCCF for Retail and Tmall) and summation operation.
- **CKML w/o MI:** To evaluate the effectiveness of multi-interest, we remove the above two modules simultaneously and use unified vectors for users and items representations.

The performance of CKML and its variants are summarized in Table 7, and we come to these conclusions:

- Comparing the performance of CKML and its first two variants, we can find that each variant brings about performance degradation when any key component is removed or replaced with other modules. This demonstrates the rationality and effectiveness of the two key designations.
- It is worthwhile noticing that CKML w/o MI achieves the worst performance on all three datasets compared to other variants with multi-interest learning. In particular, this variant has a performance decline up to **6.36%**, **15.75%**, and **26.57%** in terms of HR (**16.03%**, **23.84%**, and **29.28%** in terms of NDCG) on Yelp, Retail, and Tmall datasets. This further demonstrates the effectiveness of multi-interest for the modeling of the complex dependencies among multiple behaviors.

5.4 Study of Interests (RQ3)

We propose to explicitly separate interests into shared and specific interests to alleviate the negative impact of irrelevant interactions. To demonstrate the superiority of this correlation modeling

Table 8. Impact of Share Interests and Specific Interests

Model	Yelp		Retail		Tmall	
	HR	NDCG	HR	NDCG	HR	NDCG
CKML-Shared	0.896	0.620	0.513	0.311	0.518	0.318
CKML-Specific	0.814	0.497	0.271	0.140	0.379	0.227
CKML	**0.896**	**0.623**	**0.527**	**0.323**	**0.527**	**0.321**

Boldface denotes the highest score.

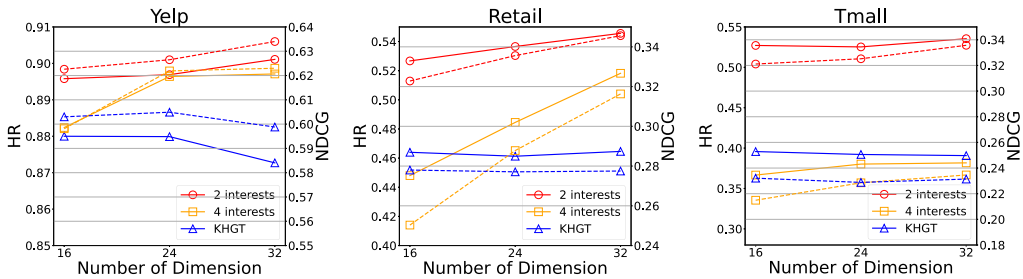

Fig. 3. Impact of the number of interests. The solid line and the dotted line represent HR and NDCG, respectively.

strategy, we replace it with two variants, namely, only shared interests and only specific interests. We keep the number of interests fixed and apply them as the basis of CKML for multi-behavior recommendation. Resulted variants are named as CKML-Shared and CKML-Specific, respectively. The results are reported in Table 8. There are some observations:

— CKML-Specific performs worse on Yelp, Retail, and Tmall datasets. This is because CKML-Specific fails to utilize information of other behaviors to assist the recommendation of target behavior as it neglects shared interests among multiple behaviors (e.g., Tip and Like on Yelp, as well as Add-to-cart and Purchase on Retail and Tmall).
— CKML, which considers shared and specific interests, achieves the best performance on all three datasets. It suggests that taking into account both share and specific interests eliminate the effect of irrelevant interactions and improve the robustness of the model.

5.5 Hyper-Parameter Study (RQ4)

5.5.1 Impact of the Number of Interests. To investigate how the number of interests affects the performance of CKML, we adjust the number of interests in the range {2,4}. For simplicity, we set the number of shared interests and specific interests to the same. The results are presented in Figure 3. We can find that when embedding size is set to 16 in line with KHGT, the model with 2 interests achieved the best results on all three datasets. Performance drops a lot when the number of interests increases from 2 to 4. Possible reason may be the too small embedding size (only 8) of each interest which can hardly learn good representations. We further extend the embedding size to 16 and 32, and we can observe that our model achieves significant performance improvement for both 2 interests and 4 interests. This verifies our above assumption. When embedding size grows larger, KHGT performs consistently worse than our model, which shows the superior performance of our proposed CKML. Moreover, KHGT has a performance drop on Yelp, Retail, and Tmall datasets when a larger embedding size is applied. Possible reason is that KHGT is easier to overfit due to the overlooking of multi-interest.

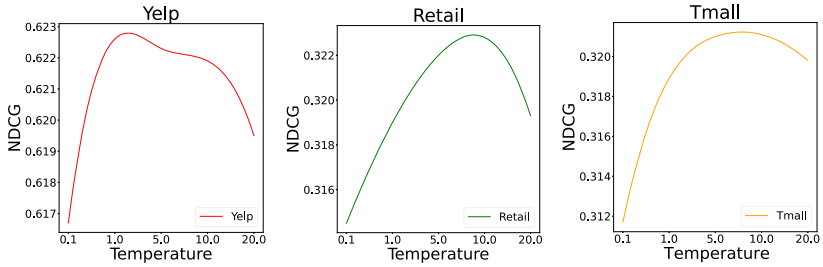

Fig. 4. Impact of temperature coefficient.

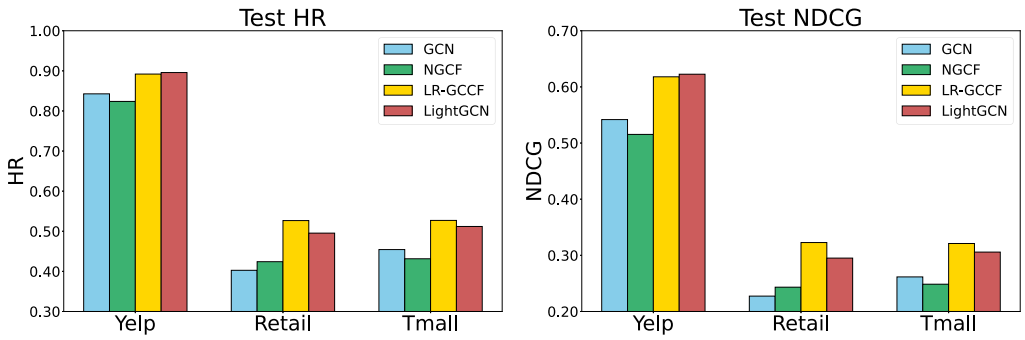

Fig. 5. Impact of GCN aggregators.

5.5.2 Impact of Temperature Coefficient. The interactions between users and items are due to a single interest or the combination of multiple interests. To investigate it, we change the temperature coefficient used for behavior allocation and the results are reported in Figure 4. We can see that a moderate temperature coefficient is needed for the CKML to achieve the best performance. And when the temperature coefficient is set too small, the performance of the model deteriorates rapidly. One possible reason is that the probability distribution of interest is close to the one-hot vector in this case, which makes it challenging to learn. Besides, the model performance degrades either if the temperature coefficient is set too large. This may be because the weights of multiple interests become similar, and the model fails to identify the interest behind the interaction well. This again illustrates the importance of exploring multiple interests.

5.5.3 Impact of GCN Aggregators. We investigate the impact of different GCN aggregators i.e., GCN [21], NGCF [39], LR-GCCF [10], and LightGCN [17]. The models with different aggregators are compared in Figure 5. We can find that LightGCN performs the best on Yelp among the four aggregators. The reason might be that removing the transformation matrix and nonlinear functions enables easier training and alleviate overfitting. CKML with LR-GCCF achieves the best performance on Retail and Tmall, probably because these two datasets contain multiple types of closely correlated behaviors, which has high requirements on the fitting ability of the model. So the introduction of a nonlinear activation function better facilitates the model to fit Retail and Tmall.

5.6 Case Study (RQ5)

5.6.1 The Visualized Analysis of Interest Initialization. We have claimed in Section 1 that initializing the clustering centers to be far apart is significant for the learning of interests. To further illustrate that the initialization process of CIE ensures that the initial centers of interest are as far

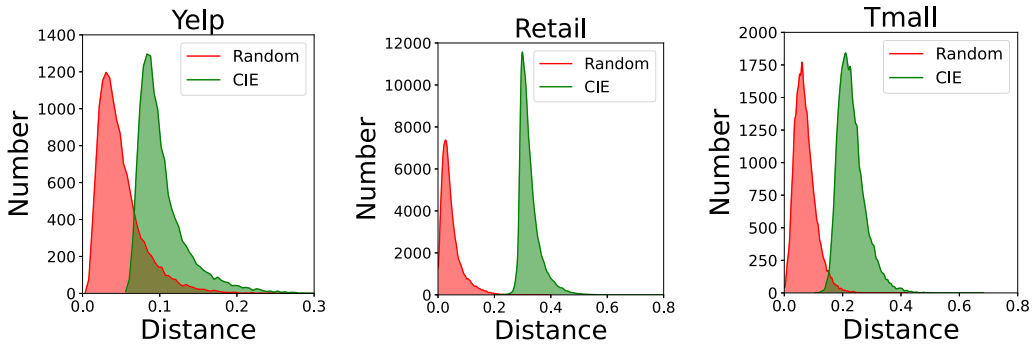

Fig. 6. The distribution of average distances.

away as possible and better than Random, we calculate the average Euclidean distance between CIE and Random on different interests of all items, then plot the distance distribution in Figure 6. Specifically, we calculate and average the Euclidean distance between all pairs of interests in $\mathbf{g}_i^k$ for each item i:

$$Distance(i) = \sum_{s=1}^{N_*} \sum_{\substack{s'=1 \\ s' \neq s}}^{N_*} \frac{\sqrt{\sum_j^{d^*} \left(\mathbf{g}_i^k[s,j] - \mathbf{g}_i^k[s',j]\right)^2}}{N_*^2 - N_*}, \tag{19}$$

where s denotes the sth interest, N_* is the number of interests. d^* is the interest embedding size. k represents the kth behavior, and here we set the target behavior as k.

We can observe the overall distribution of the average distances of interests obtained by CIE is further across all three datasets, which means the clustering centers initialized by CIE are farther apart than Random. It suggests that CIE can better initialize interest centers, enabling the model to identify the interest behind interactions efficiently.

5.6.2 The Visualized Analysis of Shared and Specific Interests. We randomly select five users and the items they have interacted with under the target behavior. In Figure 7, we visualize the representations of items under shared interest and specific interest obtained from CKML, as well as the representations obtained by KHGT.

Comparing the points with the same color in Figure 7(a)–(c), we can find that items under the shared interest and KHGT are more clustered than specific interest representations. A probable reason is that Yelp has few interactions of target behavior, which makes it hard to mine the interest-related information behind the interaction. Besides, the shared interest and KHGT introduce additional interaction information of other behaviors, which makes it better to learn the representations of items.

We further analyze Retail and Tmall, and the results of the two datasets with the same behavior (target behavior) are shown in Figures 7(d)–(f) and 7(g)–(i), respectively. We can find that items under the shared interest are more clustered than specific interest representations. One possible reason is that the strong correlation between the four behaviors (e.g., page view, favorite, cart, and purchase), which brings additional powerful interactive information to assist the learning of target behavior. Conversely, items under the shared interest representations are more closely distributed than KHGT. This is due to the possibility of CKML to better extract shared interests excluding the interference of behavior-specific interests.

5.6.3 The Visualized Analysis of Behavioral Correlation. We have depicted the explicit relevance scores ($\lambda_{k,k'}^{u,h}$ and $\lambda_{k,k'}^{i,h}$) learned by our CKML model for predicting purchases in Retail and

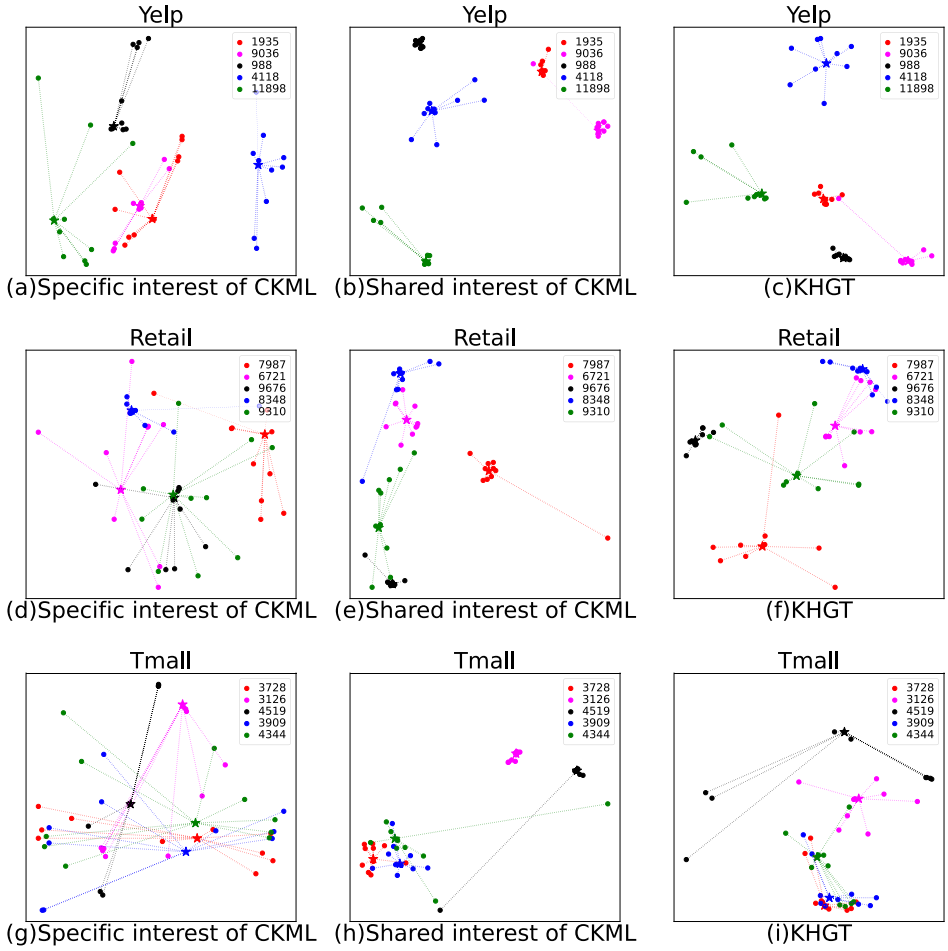

Fig. 7. Visualization of items representations via t-SNE. Points of the same color represent items being interacted with by the same user. Each star is the center of points with the same color.

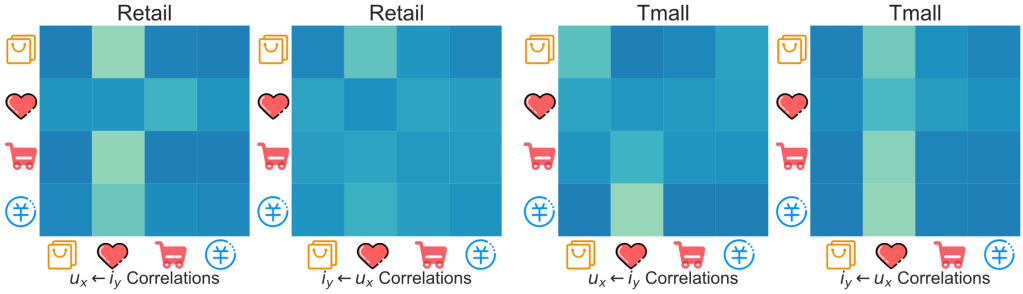

Fig. 8. Visualization of the explicit relevance learned by CKML. For each behavior on the vertical axis, we individually analyze the relevances of all behaviors with it, and express the relevances with the darkness of the square color.

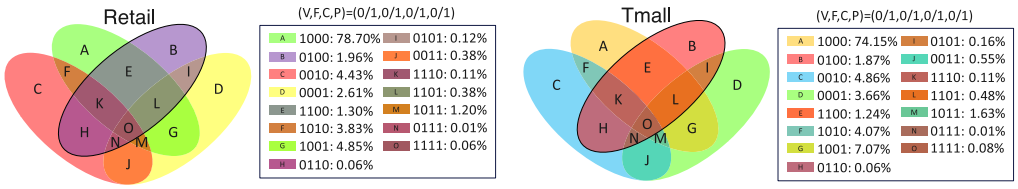

Fig. 9. Venn diagram of label correlations on the two datasets. 1/0 means have or not have this type of behavior. E.g., 0110 represents those users who only have favorite and cart behaviors with items.

Tmall datasets are in Figure 8. The visualization reveals a hierarchical and explainable correlation among different types of user-item interactions (4 types). The darkness of the colors indicates the strength of the behavioral relevance, with darker colors representing higher relevance. In each row of the figure, the squares represent the cross-type behavioral dependencies learned through our Fine-grained Behavioral Correlation. For instance, in the Retail dataset, the "purchase" behavior demonstrates higher relevance with "page view" and "cart", while exhibiting lower relevance with "favorite". Similar observations can be drawn from the results obtained from the Tmall dataset. Moreover, we find that calculating the relevance between behaviors based on aggregated information from the item side yields better discrimination. This may stem from our CKML model's ability to extract coarse-grained interests from item-item information, enabling the learning of more comprehensive behavioral correlations.

Furthermore, we analyze the label correlations to explain the above results. Figure 9 shows the behavioral label correlations with the venn diagram, where different overlaps represent different label correlations. We can find that the total proportion of X1X1 (X = 0/1) is only 0.57% and 0.73% in Retail and Tmall datasets, respectively. However, the total proportion of X0X1 (X = 0/1) is 9.04% and 12.91%, respectively. Hence, there is a limited overlap between the target behavior (purchase) and "favorites" in the two datasets, suggesting a weak correlation between these two behaviors. This observation aligns with the behavioral correlations learned by our model. The same analysis holds true for other behaviors as well.

6 CONCLUSION

In this article, we propose the CKML framework for multi-behavior recommendations. In order to make full use of knowledge-aware information to extract shared and behavior-specific interest representations, we propose the CIE module. To further learn the interest representation of each user and item under different behaviors and exchange information under different behaviors at fine granularity, we propose a GNN-based FBC module, which allocates edge weights by dynamic routing and exchanges information by self-attention mechanism. We conduct comprehensive experiments on three real-world datasets and show that the proposed CKML outperforms all state-of-the-art methods on all three datasets. Besides, the additional visualization experiment demonstrates the superiority of our well-designed shared and behavior-specific interests.

REFERENCES

[1] 2020. MindSpore. Retrieved from https://www.mindspore.cn.

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2–4, 2016*, Kimberly Keeton and Timothy Roscoe (Eds.). USENIX Association, 265–283. Retrieved from https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[3] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7–9, 2007*, Nikhil Bansal, Kirk Pruhs, and Clifford Stein (Eds.). SIAM, 1027–1035. Retrieved from http://dl.acm.org/citation.cfm?id=1283383.1283494.

[4] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. 2012. Scalable K-Means++. *Proceedings of the VLDB Endowment* 5, 7 (2012), 622–633. DOI: https://doi.org/10.14778/2180912.2180915

[5] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *Proceedings of the World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13–17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 151–161. DOI: https://doi.org/10.1145/3308558.3313705

[6] Rich Caruana. 1997. Multitask learning. *Machine Learning* 28, 1 (1997), 41–75. DOI: https://doi.org/10.1023/A:1007379606734

[7] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the KDD'20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23–27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 2942–2951. DOI: https://doi.org/10.1145/3394486.3403344

[8] Chong Chen, Weizhi Ma, Min Zhang, Zhaowei Wang, Xiuqiang He, Chenyang Wang, Yiqun Liu, and Shaoping Ma. 2021. Graph heterogeneous multi-relational recommendation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI 2021, 33rd Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The 11th Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2–9, 2021*. AAAI Press, 3958–3966. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/16515.

[9] Chong Chen, Min Zhang, Yongfeng Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020, The 32nd Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The 10th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020*. AAAI Press, 19–26. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/5329.

[10] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020, The 32nd Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The 10th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020*. AAAI Press, 27–34. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/5330.

[11] Zhiyong Cheng, Sai Han, Fan Liu, Lei Zhu, Zan Gao, and Yuxin Peng. 2023. Multi-behavior recommendation with cascading graph convolution networks. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 1181–1189. DOI: https://doi.org/10.1145/3543507.3583439

[12] 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 *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8–11, 2019*. IEEE, 1554–1557. DOI: https://doi.org/10.1109/ICDE.2019.00140

[13] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13–15, 2010 (JMLR Proceedings, Vol. 9)*, Yee Whye Teh and D. Mike Titterington (Eds.). JMLR.org, 249–256. Retrieved from http://proceedings.mlr.press/v9/glorot10a.html.

[14] 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 *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4–8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 1984–1992. DOI: https://doi.org/10.1145/3292500.3330670

[15] Wei Guo, Chang Meng, Enming Yuan, Zhicheng He, Huifeng Guo, Yingxue Zhang, Bo Chen, Yaochen Hu, Ruiming Tang, Xiu Li, and Rui Zhang. 2023. Compressed interaction graph based framework for multi-behavior recommendation. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023–4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 960–970. DOI: https://doi.org/10.1145/3543507.3583312

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016*. IEEE Computer Society, 770–778. DOI: https://doi.org/10.1109/CVPR.2016.90

[17] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 639–648. DOI: https://doi.org/10.1145/3397271.3401063

[18] Chao Huang. 2021. Recent advances in heterogeneous relation learning for recommendation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event/Montreal, Canada, 19–27 August 2021*, Zhi-Hua Zhou (Ed.). ijcai.org, 4442–4449. DOI: https://doi.org/10.24963/ijcai.2021/606

[19] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 659–668. DOI: https://doi.org/10.1145/3397271.3401072

[20] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). Retrieved from http://arxiv.org/abs/1412.6980.

[21] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*. OpenReview.net. Retrieved from https://openreview.net/forum?id=SJU4ayYgl.

[22] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37. DOI: https://doi.org/10.1109/MC.2009.263

[23] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the KDD'20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23–27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 1748–1757. DOI: https://doi.org/10.1145/3394486.3403226

[24] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3–7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 2615–2623. DOI: https://doi.org/10.1145/3357384.3357814

[25] Daryl Lim, Julian J. McAuley, and Gert R. G. Lanckriet. 2015. Top-N recommendation with missing implicit feedback. In *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16–20, 2015*, Hannes Werthner, Markus Zanker, Jennifer Golbeck, and Giovanni Semeraro (Eds.). ACM, 309–312. Retrieved from https://dl.acm.org/citation.cfm?id=2799671.

[26] Zheng Liu, Jianxun Lian, Junhan Yang, Defu Lian, and Xing Xie. 2020. Octopus: Comprehensive and elastic user representation for the generation of recommendation candidates. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 289–298. DOI: https://doi.org/10.1145/3397271.3401088

[27] Andrew McCallum, Kamal Nigam, and Lyle H. Ungar. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the 6th ACM SIGKDD International conference on Knowledge Discovery and Data Mining, Boston, MA, USA, August 20–23, 2000*, Raghu Ramakrishnan, Salvatore J. Stolfo, Roberto J. Bayardo, and Ismail Parsa (Eds.). ACM, 169–178. DOI: https://doi.org/10.1145/347090.347123

[28] Vito Claudio Ostuni, Tommaso Di Noia, Eugenio Di Sciascio, and Roberto Mirizzi. 2013. Top-N recommendations from implicit feedback leveraging linked open data. In *Proceedings of the 7th ACM Conference on Recommender Systems, RecSys'13, Hong Kong, China, October 12–16, 2013*, Qiang Yang, Irwin King, Qing Li, Pearl Pu, and George Karypis (Eds.). ACM, 85–92. DOI: https://doi.org/10.1145/2507157.2507172

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18–21, 2009*, Jeff A. Bilmes and Andrew Y. Ng (Eds.). AUAI Press, 452–461. Retrieved from https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1630&proceeding_id=25.

[30] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 3856–3866. Retrieved from https://proceedings.neurips.cc/paper/2017/hash/2cad8fa47bbef282badbb8de5374b894-Abstract.html.

[31] Noveen Sachdeva, Carole-Jean Wu, and Julian J. McAuley. 2022. On sampling collaborative filtering datasets. In *WSDM'22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event/Tempe, AZ, USA, February 21–25, 2022*, K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang (Eds.). ACM, 842–850. DOI: https://doi.org/10.1145/3488560.3498439

[32] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18–22, 2015 - Companion Volume*, Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi (Eds.). ACM, 111–112. DOI: https://doi.org/10.1145/2740908.2742726

[33] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.* 2009, Article 4 (Jan 2009), 1 pages. https://doi.org/10.1155/2009/421425

[34] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8–12, 2021*, Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 598–606. DOI: https://doi.org/10.1145/3437963.3441811

[35] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations. In *Proceedings of the RecSys 2020: 14th ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura (Eds.). ACM, 269–278. DOI: https://doi.org/10.1145/3383313.3412236

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. Retrieved from https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[37] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. Retrieved from https://openreview.net/forum?id=rJXMpikCZ.

[38] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4–8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 950–958. DOI: https://doi.org/10.1145/3292500.3330989

[39] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21–25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 165–174. DOI: https://doi.org/10.1145/3331184.3331267

[40] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the WWW'21: The Web Conference 2021, Virtual Event/Ljubljana, Slovenia, April 19–23, 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM/IW3C2, 878–887. DOI: https://doi.org/10.1145/3442381.3450133

[41] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1001–1010. DOI: https://doi.org/10.1145/3397271.3401137

[42] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive meta learning with behavior multiplicity for recommendation. In *Proceedings of the WSDM'22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event/Tempe, AZ, USA, February 21–25, 2022*, K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang (Eds.). ACM, 1120–1128. DOI: https://doi.org/10.1145/3488560.3498527

[43] Lirong Wu, Haitao Lin, Cheng Tan, Zhangyang Gao, and Stan Z. Li. 2023. Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Trans. Knowl. Data Eng.* 35, 4 (2023), 4216–4235. https://doi.org/10.1109/TKDE.2021.3131584

[44] 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 *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25–30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 2397–2406. DOI: https://doi.org/10.1145/3397271.3401445

[45] 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 *Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI 2021, 33rd Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The 11th Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2–9, 2021*. AAAI Press, 4486–4493. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/16576.

[46] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. 2021. Graph meta network for multi-behavior recommendation. In *Proceedings of the SIGIR'21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11–15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 757–766. DOI: https://doi.org/10.1145/3404835.3462972

[47] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommender systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19–25, 2017*, Carles Sierra (Ed.). ijcai.org, 3203–3209. DOI: https://doi.org/10.24963/ijcai.2017/447

[48] Mingshi Yan, Zhiyong Cheng, Chen Gao, Jing Sun, Fan Liu, Fuming Sun, and Haojie Li. 2023. Cascading residual graph convolutional network for multi-behavior recommendation. *ACM Trans. Inf. Syst.* (Mar 2023). https://doi.org/10.1145/3587693.

[49] Weifeng Zhang, Jingwen Mao, Yi Cao, and Congfu Xu. 2020. Multiplex graph neural networks for multi-behavior recommendation. In *Proceedings of the CIKM'20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19–23, 2020*. Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 2313–2316. DOI: https://doi.org/10.1145/3340531.3412119

Received 11 November 2022; revised 27 May 2023; accepted 25 June 2023