

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

Chang Meng^{*1}, Ziqi Zhao², Wei Guo³, Yingxue Zhang⁴,

Haolun Wu⁵, Chen Gao⁶, Dong Li³, Xiu Li¹, Ruiming Tang³

¹Shenzhen International Graduate School, Tsinghua University ²Harbin Engineering University

³Noah's Ark Lab, Huawei ⁴Huawei Technologies Canada

⁵McGill University ⁶Department of Electronic Engineering, Tsinghua University

mengc21@mails.tsinghua.edu.cn, zziqi@hrbeu.edu.cn

{guowei67, yingxue.zhang, lidong106, tangruiming}@huawei.com

haolun.wu@mail.mcgill.ca, chgao96@gmail.com, li.xiu@sz.tsinghua.edu.cn

ABSTRACT

Multi-types of behaviors (e.g., clicking, adding to cart, 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. Additionally, we use the self-attention mechanism to correlate different behavioral information at the interest level. Empirical results on three real-world datasets verify the effectiveness and efficiency of our model in exploiting multi-behavior data. Further experiments demonstrate the effectiveness of each module and the robustness and superiority of the shared and specific modelling paradigm for multi-behavior data.

1 INTRODUCTION

Collaboration filtering (CF) [32] 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, 23, 38, 45] 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 purchasing in the e-commerce scenario. As various behaviors may

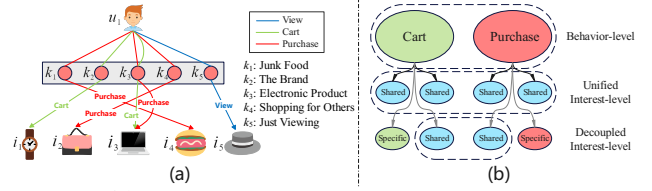


Figure 1: (a) An example of multi-faceted interests behind multiple behaviors on an e-commerce scenario. (b) An example of behavior correlation at different granularities. Black and gray arrows represent different interest divisions, and dashed boxes represent correlations of information.

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.** For example, DIPN [14] and MATN [42] apply the attention network and transformer network separately to model dependence between different behaviors. Further, MBGCN [19] and KHGT [43] strengthen multi-behavior data with graph neural network (GNN) [21, 22, 36] **by regarding multiple behaviors data as heterogeneous graphs to learn more complex heterogeneous information behind the interaction between users and items.** The second category utilizes multi-behavior data with multi-task learning (MTL) [6]. For instance, NMTR [12] leverages all behaviors of users as prediction targets to improve the learning of users and items. EHCF [10] and GHCF [9] then improve NMTR [12] with knowledge transfer and GNN to correlate the predictions, respectively.

However, these multi-behavior recommendation methods ignore the multi-faceted interests behind different behaviors. As shown in Figure 1(a), here we exhibit a toy example, a user (u_1) generally adopts different items (i_1, i_2, \dots, i_5) under different behaviors. **Each behavior belongs to a plurality of different interests (k_1, k_2, \dots, k_5), and each interaction is based on one or more interests.** We can observe that different interests under the same behaviors may have different effects on the prediction of target behaviors. For example, interests k_2 and k_3 are both under the behavior of carting. **However, for predicting whether the user will purchase i_3 , interest k_3 is more effective and meaningful, because the user add i_3 to cart based on k_3 and add the less relevant item i_1 to cart based on k_2 .** And we can also find that some interests are specific only to a certain behavior

^{*} Work done when he is a research intern at Noah's Ark Lab, Huawei.

(e.g. user u_1 adds item i_1 to cart only because of interest k_2), and these 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.

Some recent works have attempted to leverage multi-interest learning for recommendation, either utilize behavioral collaborative signals to learn multi-interest representations, or leverage structured relational information to construct multi-interest representations. For the approaches fall under the first category, MIND [24], ComiRec [7] and DGCF [40] implicitly cluster historical user interactions by using powerful encoders, like dynamic routing and self-attention. For the latter category, KGIN [39] and KTUP [5] seek to leverage the auxiliary semantic information of knowledge graph to model multi-interests. Despite the effectiveness of these methods, they are all designed for single behavior, which share two common limitations when applied for multi-behavior recommendation:

- **Suboptimal Correlation Modeling.** Existing multi-interest methods are designed for single behavior, all of which adopts a group of unified interests for each user. However, as illustrated above, **different interests under the same behaviors may have different effects on the prediction of target behaviors**, and some interests are specific only to one behavior. **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-level*, as shown in Figure 1 (b). The *unified interest-level* roughly correlate the divided shared interests of one behavior with the ones of other behaviors, which will lead to a suboptimal modeling of correlation. So, a more reasonable strategy is necessary to better model behavior correlations.
- **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 [28]. Existing multi-interest methods like MIND [24] and DGCF [40] initialize clustering centers with random vectors, which lead to sub-optimal results as the generated centers may be very close to each other. **Besides the initialization of clustering centers, the process of clustering via collaborative signals are also important for the clustering results.** Methods like KGIN [39] and KTUP [5] utilize the semantic information from knowledge graph to learn interest representation. However, they are possibly sub-optimal either, as 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 center 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 **Coarse-to-fine Knowledge-enhanced Multi-interest Learning** framework (CKML). To handle the first limitation, CKML decouples interests into shared and behavior-specific parts for each behavior, then model the behavior correlation under the *decoupled interest-level*, 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: **Coarse-grained Interest Extracting (CIE)** module and **Fine-grained Behavioral Correlation (FBC)** module. For the first module, **it firstly learn representations for interests under each behavior with the paradigm of graph neural networks** (capture the knowledge-aware relations between items), and **then use the representations of knowledge-aware relations to initiate shared and behavior-specific interests for every behavior.** For the second module, we design a GNN-based framework (capture the high-order relations between users and items) with dynamic routing mechanism [30] to assign every edge on each graph (means user-item interaction under a specific behavior) 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 apply the self-attention mechanism on these interest representations to model the *decoupled interest-level* correlation between different behaviors for multi-behavior prediction.

To summarize, this work makes the following contributions:

- We propose a novel Coarse-to-fine Knowledge-enhanced Multi-interest Learning framework (CKML) 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 Coarse-grained Interest Extracting (CIE) module and a Fine-grained Behavioral Correlation (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

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 [42] then uses transformer to encode the interactions of multiple behaviors. Further, the GNN-based methods like MGNN [49], MBGCN [19], GHCF [9] and MBGMN [44] propose to leverage message passing on graphs to model high-order multi-behavioral interactive information. Moreover, KHGT [43]

combines GNN and transformer together to model the global behavioral information, which not only captures the higher-order behavior between nodes, but also addresses the dynamics of behavior. The other category is to model different behaviors with MTL. DIPN [14], MGNN [49] and GHCF [9] regard the aggregated representations of different behaviors as shared input and simply use aggregated representations to predict each behavior individually. NMTR [12], EHCF [10] and MBGMN [44] 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.

2.2 Multi-interest Recommendation

Existing methods for multi-interest learning can be divided into two paradigms. **One paradigm is utilizing behavioral collaborative signals to learn multi-interest representations.** For example, MIND [24] applies a dynamic routing mechanism to assign each interaction to interests. On this basis, Comirec [7] leverages self-attentive mechanism to extract user interests. To better learn interest representations, SINE [33] and Octopus [27] 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 [40] introduces the dynamic routing mechanism into graphs with independence modeling among interests for multi-interest learning. The other paradigm is leveraging structured relational information to construct multi-interest representations. For instance, KGIN [39] exploits knowledge graph’s structural information to learn the representations of different interests. KTUP [5] rises a new translation-based model, which leverages implicit interests to capture the relationship between users and items. 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

Multi-Behavior Interaction Graph. Let $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$ represent the set of users and $\mathcal{I} = \{i_1, i_2, \dots, i_N\}$ 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 $\mathcal{Y}_{u-i} = \{Y_{u-i}^1, Y_{u-i}^2, \dots, Y_{u-i}^K\}$, where Y_{u-i}^k represents the interaction matrix of behavior k , $y_{ui}^k = 1$ denotes that user u interacts with item i under behavior k , otherwise $y_{ui}^k = 0$. 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, $\mathcal{E}_{u-i} = \cup_{k \in \mathcal{R}_{u-i}} \mathcal{E}_{u-i}^k$ 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. $\mathcal{A}_{u-i} = \cup_{k \in \mathcal{R}_{u-i}} \mathcal{A}_{u-i}^k$ is the adjacency matrix set with \mathcal{A}_{u-i}^k denoting adjacency matrix of a specific behavior graph $\mathcal{G}_{u-i}^k = (\mathcal{V}, \mathcal{E}_{u-i}^k, \mathcal{A}_{u-i}^k)$.

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 $\mathcal{A}_{i-i} = \{\mathcal{A}_{i-i}^1, \mathcal{A}_{i-i}^2, \dots, \mathcal{A}_{i-i}^{|\mathcal{R}_{i-i}|}\}$, which can be constructed with different reasons, such as items with same category, from the same restaurant, or interacted by similar users.

Task Description. Generally, there is a target behavior to be optimized (e.g., purchase), which we denote as Y_{u-i}^K , and other behaviors $\{Y_{u-i}^1, Y_{u-i}^2, \dots, Y_{u-i}^{K-1}\}$ (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) Coarse-grained Interest Extracting (CIE) module, which utilizes knowledge-aware relations to extract shared and behavior-specific interests for multiple behaviors; and (2) Fine-grained Behavioral Correlation (FBC) module, which allocates 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:

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

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, 39, 43]. Inspired by the strong semantics of relations in the knowledge-aware relation graph [37, 39, 43], we propose a Coarse-grained Interest Extraction (CIE) module to **extract users’ interests which motivates users’ interactions of multiple behaviors.** In this way, we obtain the initial interest clustering centers. 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.**

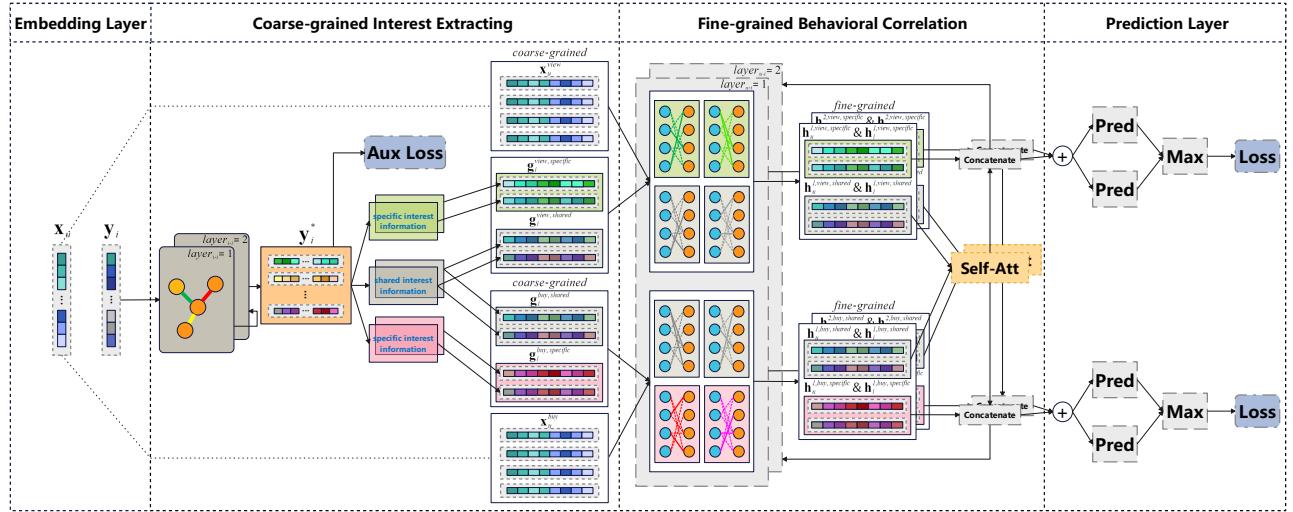


Figure 2: Illustration of the proposed CKML. For brevity, only two behaviors are represented here. The green and red rectangles represent the behavior-specific interests, while the grey rectangle represents the shared interests. The orange circles represent items, while the blue circles represent users.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

4.2.1 Knowledge-aware Relation Modeling. Most existing multi-interest methods initialize interests with random generated vectors [24, 40], 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, 39, 43], 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 $\mathcal{G}_{i-i}^1, \mathcal{G}_{i-i}^2, \dots, \mathcal{G}_{i-i}^{|\mathcal{R}_{i-i}|}$, and the corresponding adjacency matrices are $A_{i-i}^1, A_{i-i}^2, \dots, A_{i-i}^{|\mathcal{R}_{i-i}|}$. As for the message propagation, we adopt the state-of-the-art GCN models, such as LightGCN [17], LR-GCCF [11], GCN [21] and NGCF [38], for graph information aggregation. And the neighbor propagation process in each layer of each sub-graph can be formulated as:

$$y_i^{r,l} = \text{Agg} \left(y_j^{r,l-1}, A_{i-i}^r \right) \quad (2)$$

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

$$y_i^r = \sum_{l=0}^{L_{i-i}} y_i^{r,l} / (L_{i-i} + 1) \quad (3)$$

where $y_i^r \in \mathbb{R}^{1 \times d}$ 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 [34] 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:

$$y_i^* = \text{Concatenate}_{r \in \mathcal{R}_{i-i}} y_i^r, \quad (4)$$

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:

$$g_i^{k,spe} = \text{Concatenate}_{s=1}^{N_{spe}} \left(\text{LeakyReLU} \left(y_i^* \cdot W_s^k + b_s^k \right) \right) \quad (5)$$

where N_{spe} is the number of specific interests for each behavior, s is the s -th interest, $W_s^k \in \mathbb{R}^{(|\mathcal{R}_{i-i}| \times d) \times (\frac{d}{N_{spe}})}$ and $b_s^k \in \mathbb{R}^{1 \times (\frac{d}{N_{spe}})}$ are transformation matrix and bias matrix, and $g_i^{k,spe}$ 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:

$$g_i^{k,sha} = \text{Concatenate}_{s=1}^{N_{sha}} \left(\text{LeakyReLU} \left(y_i^* \cdot W_s + b_s \right) \right) \quad (6)$$

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

Algorithm 1: The allocation of interests for the k -th behavior on the l -th layer.

Input : $A_{u-i}^k \in \mathbb{R}^{(M+N) \times (M+N)}$, $\forall u \in \mathcal{U}, \forall i \in \mathcal{I}$:
 $\mathbf{x}_u^{l,k}, \mathbf{g}_i^{l,k}, \mathbf{t}_u^k, \mathbf{t}_i^k \in \mathbb{R}^{N_s \times d^*}$

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

```

/* Step 1. Initialize. */
1  $\mathcal{E}_{u-i}^k \leftarrow A_{u-i}^k, \mathbf{a}_0^{l,k} \leftarrow \mathbf{1}^{N_s \times |\mathcal{E}_{u-i}^k|}$ 
2  $\mathbf{h}_{u,0}^{l,k} \leftarrow \mathbf{x}_u^{l,k} + \mathbf{t}_u^k, \mathbf{h}_{i,0}^{l,k} \leftarrow \mathbf{g}_i^{l,k} + \mathbf{t}_i^k$ 
3 for  $t = 1$  to  $N_{iter}$  do
  /* Step 2. Normalize coefficients. */
  4  $\mathbf{a}_t^{l,k} \leftarrow \text{Softmax}(\mathbf{a}_{t-1}^{l,k} / \tau)$ 
  5 for  $s = 1$  to  $N_s$  do
    /* Step 3. Embedding propagation. */
    6  $\mathbf{h}_{u,t}^{l,k}[s] \leftarrow \sum_{(u,i) \in \mathcal{E}_{u-i}^k} \frac{\mathbf{a}_t^{l,k}[s, (u,i)] \cdot \mathbf{h}_{i,0}^{l,k}[s]}{\sum_{(u,i) \in \mathcal{E}_{u-i}^k} \mathbf{a}_t^{l,k}[s, (u,i)]}, \forall u \in \mathcal{U}$ 
    7  $\mathbf{h}_{i,t}^{l,k}[s] \leftarrow \sum_{(i,u) \in \mathcal{E}_{u-i}^k} \frac{\mathbf{a}_t^{l,k}[s, (i,u)] \cdot \mathbf{h}_{u,0}^{l,k}[s]}{\sum_{(i,u) \in \mathcal{E}_{u-i}^k} \mathbf{a}_t^{l,k}[s, (i,u)]}, \forall i \in \mathcal{I}$ 
    /* Step 4. Update coefficients. */
    8  $\mathbf{a}_t^{l,k}[s] \leftarrow \mathbf{a}_{t-1}^{l,k}[s] +$ 
    9  $\text{Stack}_{(p,q) \in \mathcal{E}_{u-i}^k} \left( \frac{\mathbf{h}_{q,0}^{l,k}[s]}{\|\mathbf{h}_{q,0}^{l,k}[s]\|} \cdot \tanh\left(\frac{\mathbf{h}_{p,t}^{l,k}[s]}{\|\mathbf{h}_{p,t}^{l,k}[s]\|}\right)^\top \right)$ 
10 end
11 end
/* Step 5. Information aggregation. */
12  $\mathbf{h}_u^{l,k} \leftarrow \text{Agg}(\text{Concatenate}_{s=1}^{N_s}(\mathbf{h}_{u,N_{iter}}^{l,k}[s]), A_{u-i}^k), \forall u \in \mathcal{U}$ 
13  $\mathbf{h}_i^{l,k} \leftarrow \text{Agg}(\text{Concatenate}_{s=1}^{N_s}(\mathbf{h}_{i,N_{iter}}^{l,k}[s]), A_{u-i}^k), \forall i \in \mathcal{I}$ 

```

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

$$\mathbf{g}_i^k = \mathbf{g}_i^{k, spe} \parallel \mathbf{g}_i^{k, sha} \quad (7)$$

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

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 Fine-grained Behavioral Correlation (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, 40] for behavior allocation. We firstly partition the provided multi-behavior user-item graph \mathcal{G}_{u-i} into behavior-specific sub-graphs $\mathcal{G}_{u-i}^1, \mathcal{G}_{u-i}^2, \dots, \mathcal{G}_{u-i}^K$, and the corresponding adjacency matrices are $A_{u-i}^1, A_{u-i}^2, \dots, A_{u-i}^K$, which can be formulated as:

$$A_{u-i}^k = \begin{pmatrix} 0 & \mathbf{Y}_{u-i}^k \\ (\mathbf{Y}_{u-i}^k)^T & 0 \end{pmatrix} \quad (8)$$

where \mathbf{Y}_{u-i}^k is the user-item adjacency interaction matrix of behavior k , $A_{u-i}^k \in \mathbb{R}^{(M+N) \times (M+N)}$, M and N denote the number of users and items, respectively. As for the processing of time, we simply follow KHGT [43], and initialize the embedding of time as $\mathbf{t}_{u-i}^k \in \mathbb{R}^{N_s \times d^*}$ for each interaction.

To better illustrate the process of the allocation of interests on each layer, we take the k -th behavior on the l -th layer as an example. As shown in Algorithm 1, we use $\mathcal{E}_{u-i}^k = \{(p, q) | A_{u-i}^k[p, q] \neq 0\}$ 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 (Step 1). And for detail, we have $\mathbf{x}_u^{0,k} = \mathbf{x}_u^k$ and $\mathbf{g}_i^{0,k} = \mathbf{g}_i^k$. Next, we start iterative process. In the t -th iteration, in order to get distributions across all interests, we use the softmax function to normalize these coefficients (Step 2):

$$\mathbf{a}_t^{l,k}(s, (u, i)) = \frac{\exp(\mathbf{a}_{t-1}^{l,k}(s, (u, i)) / \tau)}{\sum_{s=1}^{N_s} \exp(\mathbf{a}_{t-1}^{l,k}(s, (u, i)) / \tau)} \quad (9)$$

where $\mathbf{a}_t^{l,k}$ denotes the vector of weight coefficients of each edge on the l -th layer of graph \mathcal{G}_{u-i}^k in the t -th iteration, τ is the temperature coefficient, $\forall (u, i) \in \mathcal{E}_{u-i}^k$, s denotes the s -th 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, $\mathbf{h}_{u,t}^{l,k}[s]$ and $\mathbf{h}_{i,t}^{l,k}[s]$ represent the s -th 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, $\mathbf{a}_t[s]$ denotes the updated weight of edges at the t -th iteration of the s -th interest. 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 each node on each layer, 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 [35] because the behavior-specific interests contain few useful information for the target behavior and may contain noise. For instance, in the Yelp and MovieLens-10M datasets, 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:

$$\left\{ \begin{aligned} \tilde{\mathbf{h}}_n^{l,k,sha} &= \text{MH} - \text{Att}(\mathbf{h}_n^{l,k,sha}) + \sum_{k'=1}^K \mathbf{h}_n^{l,k',sha} \\ \text{MH} - \text{Att}(\mathbf{h}_n^{l,k,sha}) &= \text{Concatenate}(\sum_{h=1}^H \sum_{k'=1}^K \lambda_{k,k'}^{n,h} \cdot \tilde{\mathbf{v}}^h \cdot \mathbf{h}_n^{l,k',sha}) \\ \lambda_{k,k'}^{n,h} &= \frac{\exp \tilde{\lambda}_{k,k'}^{n,h}}{\sum_{k'=1}^K \exp \tilde{\lambda}_{k,k'}^{n,h}} \\ \tilde{\lambda}_{k,k'}^{n,h} &= \frac{(\tilde{\mathbf{Q}}^h \cdot \mathbf{h}_n^{l,k,sha})^\top (\tilde{\mathbf{K}}^h \cdot \mathbf{h}_n^{l,k',sha})}{\sqrt{d^*/H}} \end{aligned} \right. \quad (10)$$

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. $n \in \mathcal{U} \cup \mathcal{I}$ denotes the user u or item i , $\lambda_{k,k'}^{n,h}$ represents the relevance score between k -th and k' -th behaviors of the h -th head for the node n . For the information propagation of the k -th behavior, we have:

$$\left\{ \begin{aligned} \mathbf{x}_u^{l+1,k} &= \mathbf{h}_u^{l,k,spe} \parallel \tilde{\mathbf{h}}_u^{l,k,sha} + \mathbf{x}_u^{l,k}, \forall u \in \mathcal{U} \\ \mathbf{g}_i^{l+1,k} &= \mathbf{h}_i^{l,k,spe} \parallel \tilde{\mathbf{h}}_i^{l,k,sha} + \mathbf{g}_i^{l,k}, \forall i \in \mathcal{I} \end{aligned} \right. \quad (11)$$

where $l \in [0, 1, \dots, L_{u-i}]$, L_{u-i} is the number of GNN layer, (\parallel) is the concatenated operation for two vectors.

4.4 Joint Optimization

4.4.1 The Prediction of the U-I Interaction. In the above parts, we have obtained the output $\mathbf{h}_u^{l,k,spe}$ and $\tilde{\mathbf{h}}_u^{l,k,sha}$, $\forall l \in [1, 2, \dots, L_{u-i}], \forall k \in [1, 2, \dots, K], \forall u \in \mathcal{U}$, similar operations are applied for the item i . To aggregate the information of each layer, we follow KHGT [43], and simply add them up. The unified formulation is:

$$\mathbf{h}_v^{*,k} = \sum_{l=1}^{L_{u-i}} (\mathbf{h}_v^{l,k,spe} \parallel \tilde{\mathbf{h}}_v^{l,k,sha}), \forall v \in \mathcal{U} \cup \mathcal{I} \quad (12)$$

where $\mathbf{h}_v^{*,k} \in \mathbb{R}^{N_s \times d^*}$, k represents the k -th 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:

$$\hat{\mathbf{o}}_{u,i}^k = \max_{s=1}^{N_s} (\sum_j^{d^*} (\mathbf{h}_u^{*,k}[s] \circ \mathbf{h}_i^{*,k}[s])[j]) \quad (13)$$

where $s \in [1, 2, \dots, N_s]$ denotes the s -th interest, (\circ) is the hadamard product operation.

Finally, to perform the model optimization, we follow KHGT [43] and use marginal pair-wise *Bayesian Personalized Ranking* (BPR) loss function to minimize the following loss function:

$$\mathcal{L}_{u-i} = \sum_{k=1}^K \sum_{(u,p,q) \in O_{u-i,k}} \alpha^k * \max(0, 1 - \hat{\mathbf{o}}_{u,p}^k + \hat{\mathbf{o}}_{u,q}^k) \quad (14)$$

where the $\alpha^k \in [0, 1]$ denotes the coefficient of loss for each 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 [41], we use the information of item-item relations to reconstruct the item-item graphs, which can be considered as a self-supervised relation reconstruction 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{y}_i^* at Section 4.2.1, we calculate prediction scores for each relation between items:

$$\hat{\mathbf{o}}_{i,i'}^r = \sum_j^d (\mathbf{y}_i^* \circ \mathbf{y}_{i'}^*)[j] \quad (15)$$

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

$$\mathcal{L}_{i-i} = - \sum_{r=1}^{|\mathcal{R}_{i-i}|} \sum_{(i,p,q) \in O_{i-i,r}} \ln \sigma(\hat{\mathbf{o}}_{i,p}^r - \hat{\mathbf{o}}_{i,q}^r) \quad (16)$$

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:

$$\mathcal{L}_{total} = \mathcal{L}_{u-i} + \beta \mathcal{L}_{i-i} + \lambda \|\Theta\|_F^2 \quad (17)$$

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 $O(K L_{u-i} d^2)$, 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 experiment to validate it in Section 5.2.

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 matrix 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.

Table 1: 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}
MovieLens	67,788	8,704	9.9×10^6	4,970,984	{Dislike, Neutral, Like}
Online Retail	147,894	99,037	7.7×10^6	642,916	{Page View, Favorite, Cart, Purchase}

5 EXPERIMENTS

5.1 Experimental Setting

5.1.1 Dataset Description. Closely following KHGT¹ [43], we evaluate our model on three datasets (i.e., **Yelp**, **MovieLens-10M**, and **Online Retail**) with the same parameter settings and preprocessing as KHGT. The behavior types and statistics of the three datasets are shown in Table 1.

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 and we set $N = 10$ by default in all experiments. To fairly compare our models and baselines, we follow the evaluation settings of KHGT. Specifically, 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.

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 (BiasMF [23], DMF [45], AutoRec [31]); (B) Single-behavior graph models (ST-GCN [48], NGCF² [38], DGCF [40], KGAT [37]); (C) Multi-behavior non-graph models (NMTR [12], DIPN [14], MATN[42]); (D) Multi-behavior graph models (NGCF_M [38], LightGCN_M³ [17], MBGCN [19], KHGT [43]).

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 Sections 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

5.2.1 Effectiveness Comparison. Table 2 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%**, **6.67%** and **11.65%** in terms of HR (**3.48%**, **14.91%** and **16.19%** in terms of

Table 2: The overall performance comparison. Boldface denotes the highest score and underline indicates the results of the best baselines. ★ represents significance level p -value < 0.05 of comparing CKML with the best baseline.

Model	Yelp		MovieLens		Retail	
	HR	NDCG	HR	NDCG	HR	NDCG
BiasMF	0.755	0.481	0.767	0.490	0.262	0.153
DMF	0.756	0.485	0.779	0.485	0.305	0.189
AutoRec	0.765	0.472	0.658	0.392	0.313	0.190
ST-GCN	0.775	0.465	0.738	0.444	0.347	0.206
NGCF	0.789	0.500	0.790	0.508	0.302	0.185
DGCF	0.861	0.587	0.827	0.499	0.194	0.101
KGAT	0.835	0.543	0.817	0.514	0.377	0.214
NMTR	0.790	0.478	0.808	0.531	0.332	0.179
DIPN	0.791	0.500	0.811	0.540	0.317	0.178
MATN	0.826	0.530	0.847	0.569	0.354	0.209
NGCF _M	0.793	0.492	0.825	0.546	0.374	0.221
LightGCN _M	0.873	0.573	<u>0.869</u>	0.595	<u>0.472</u>	0.277
MBGCN	0.796	0.502	0.826	0.553	0.369	0.222
KHGT	<u>0.880</u>	<u>0.603</u>	0.861	<u>0.597</u>	0.464	<u>0.278</u>
CKML	0.896★	0.624★	0.927★	0.686★	0.527★	0.323★
Rel Impr.	1.82%	3.48%	6.67%	14.91%	11.65%	16.19%

NDCG) on Yelp, MovieLens and Retail 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 DMF in most datasets and metrics at the single-behavior and multi-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. However, special designs are required to better utilize the graph information and multi-behavior information, and naive application may even drop the performance. For example, NGCF_M, which uses multi-behavior information, perform much worse than MATN.

CKML consistently outperforms GNN based multi-behavior baseline models. Our proposed CKML surpasses in the performance of NGCF_M, LightGCN_M, MBGCN, and the state-of-the-art multi-behavior model KHGT. 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.

Table 3: Training time comparison (seconds per epoch) of different methods on all three datasets.

Model	Training time (s)	Dataset		
		Yelp	MovieLens	Retail
KHGT		46.65	56.11	74.31
CKML		40.29	44.07	59.88

¹<https://github.com/akaxlh/KHGT>

²https://github.com/xiangwang1223/neural_graph_collaborative_filtering

³<https://github.com/kuandeng/LightGCN>

5.2.2 Efficiency Comparison. In addition to effectiveness, efficiency is also important. Table 3 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%**, **21.46%**, and **19.42%** 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.

Table 4: Performance of different CKML variants. The significance level p -value for all data in the table is < 0.05 .

Model	Yelp		MovieLens		Retail	
	HR	NDCG	HR	NDCG	HR	NDCG
CKML	0.896	0.624	0.927	0.686	0.527	0.323
CKML w/o CIE	0.893	0.619	0.904	0.629	0.510	0.310
CKML w/o FBC	0.887	0.610	0.884	0.598	0.491	0.290
CKML w/o MI	0.839	0.524	0.856	0.580	0.444	0.246

5.3 Ablation Study

CKML is built with several important designations including the Multi-Interest (MI), the Coarse-grained Interest Extracting (CIE) and the Fine-grained Behavioral Correlation (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 (LighGCN for Yelp and MovieLens, GCCF for Retail) 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 4. We can obtain the following observations:

- Comparing the performance of CKML and its first three 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 three 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 16.03%, 15.45% and 23.84% in terms of NDCG on Yelp, MovieLens and Retail datasets. This further demonstrates the effectiveness of multi-interest for the modeling of the complex dependencies among multiple behaviors.

Table 5: Impact of share interests and specific interests

Model	Yelp		MovieLens		Retail	
	HR	NDCG	HR	NDCG	HR	NDCG
CKML-Shared	0.896	0.620	0.834	0.546	0.513	0.311
CKML-Specific	0.814	0.497	0.925	0.680	0.271	0.140
CKML	0.896	0.624	0.927	0.686	0.527	0.323

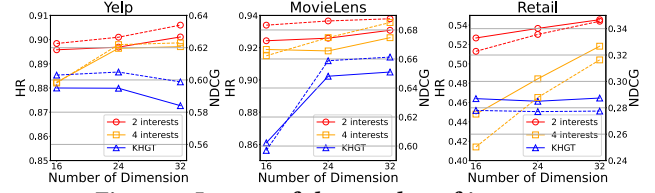


Figure 3: Impact of the number of interests

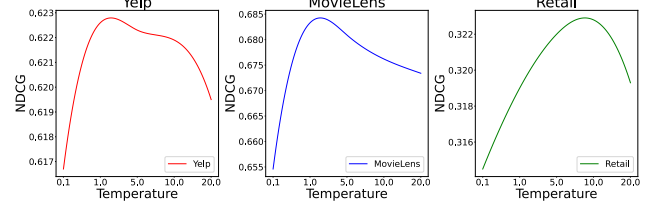


Figure 4: Impact of temperature coefficient

5.4 Study of Interests

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 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 5. There are some observations:

- We can find that CKML-Shared achieves poor performance on MovieLens dataset. Possible reason is that the behavior types in this dataset are Like, Neutral and Dislike, which are mutually exclusive with each other. So CKML-Shared may suffers from the negative impact of irrelevant interactions from other behaviors on MovieLens dataset. Besides, we also find that there exists no intersection of interaction data between different behaviors on MovieLens, that is, a user can interact with the same item through at most one type of behavior. This further validates our assumption.
- CKML-Specific performs worse on Yelp and Retail 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).
- 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

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

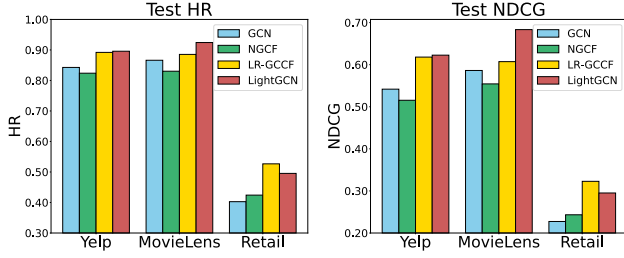


Figure 5: Impact of GCN aggregators.

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 and Retail dataset when a larger embedding size is applied. Possible reason is that KHGT is easier to overfit due to the overlooking of multi-interest.

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 GNN aggregators. We investigate the impact of different GNN aggregators i.e. GCN [21], NGCF [38], LR-GCCF [11] and LightGCN [17]. The models with different aggregators are compared in Figure 5. We can find that LightGCN performs the best on Yelp and MovieLens 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, probably because Retail contains 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.

5.6 Case Study

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 validate that the initialized representations of CIE are better than Random, we

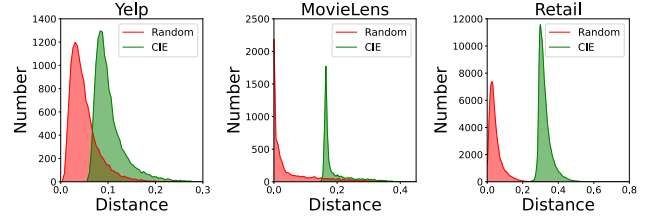


Figure 6: The distribution of average distances.

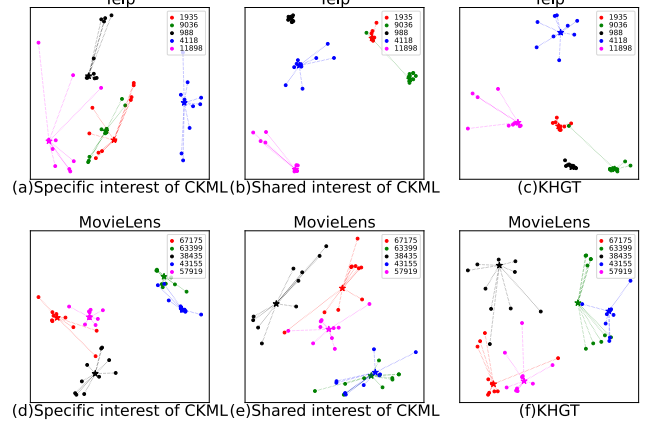


Figure 7: Visualization of items representations via t-SNE. Points of the same color represent items being interacted with by the same user. Each star is the center of points with the same color.

calculate the average Euclidean distance among different interests on all items for CIE and random, then plot the distance distribution in Figure 6. 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 visualization 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)(b)(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. By comparing Figure 7(d)(e)(f), we can observe that the result of specific interest of CKML exhibit much tighter clusters, showing a significant grouping effect for items that have been purchased by users. While the clustering effect of the other two is less pronounced. It is possibly due to the weak correlation between the three behaviors (e.g., like, neutral and dislike) on MovieLens dataset, which brings noise to the representation of shared interests. In all, the results illustrate the

effectiveness of introducing specific interest on datasets which is similar to MovieLens.

6 CONCLUSION

In this paper, 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.

ACKNOWLEDGMENTS

We thank MindSpore [1] for the partial support of this work, which is a new deep learning computing framework.

REFERENCES

- [1] MindSpore. In <https://www.mindspore.cn> (2020).
- [2] ABADI, M., BARHAM, P., CHEN, J., CHEN, Z., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., IRVING, G., ISARD, M., ET AL. {TensorFlow}: A system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (2016), pp. 265–283.
- [3] ARTHUR, D., AND VASSILVITSKII, S. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007* (2007), N. Bansal, K. Pruhs, and C. Stein, Eds., SIAM, pp. 1027–1035.
- [4] BAHMANI, B., MOSELEY, B., VATTANI, A., KUMAR, R., AND VASSILVITSKII, S. Scalable k-means++. *Proc. VLDB Endow.* 5, 7 (2012), 622–633.
- [5] CAO, Y., WANG, X., HE, X., HU, Z., AND CHUA, T. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019* (2019), L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, and L. Zia, Eds., ACM, pp. 151–161.
- [6] CARUANA, R. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.
- [7] CEN, Y., ZHANG, J., ZOU, X., ZHOU, C., YANG, H., AND TANG, J. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), pp. 2942–2951.
- [8] CEN, Y., ZHANG, J., ZOU, X., ZHOU, C., YANG, H., AND TANG, J. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), pp. 2942–2951.
- [9] CHEN, C., MA, W., ZHANG, M., WANG, Z., HE, X., WANG, C., LIU, Y., AND MA, S. Graph heterogeneous multi-relational recommendation. In *AAAI* (2021), vol. 35, pp. 3958–3966.
- [10] CHEN, C., ZHANG, M., ZHANG, Y., MA, W., LIU, Y., AND MA, S. Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 19–26.
- [11] CHEN, L., WU, L., HONG, R., ZHANG, K., AND WANG, M. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *AAAI* (2020), vol. 34, pp. 27–34.
- [12] GAO, C., HE, X., GAN, D., CHEN, X., FENG, F., LI, Y., CHUA, T.-S., AND JIN, D. Neural multi-task recommendation from multi-behavior data. In *ICDE* (2019), IEEE, pp. 1554–1557.
- [13] GLOROT, X., AND BENGIO, Y. Understanding the difficulty of training deep feed-forward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (2010), JMLR Workshop and Conference Proceedings, pp. 249–256.
- [14] GUO, L., HUA, L., JIA, R., ZHAO, B., WANG, X., AND CUI, B. Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior. In *SIGKDD* (2019), pp. 1984–1992.
- [15] GUO, W., SU, R., TAN, R., GUO, H., ZHANG, Y., LIU, Z., TANG, R., AND HE, X. Dual graph enhanced embedding neural network for CTR prediction. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021* (2021), F. Zhu, B. C. Ooi, and C. Miao, Eds., ACM, pp. 496–504.
- [16] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. In *CVPR* (2016), pp. 770–778.
- [17] HE, X., DENG, K., WANG, X., LI, Y., ZHANG, Y., AND WANG, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. *arXiv preprint arXiv:2002.02126* (2020).
- [18] HUANG, C. Recent advances in heterogeneous relation learning for recommendation. *arXiv preprint arXiv:2110.03455* (2021).
- [19] JIN, B., GAO, C., HE, X., JIN, D., AND LI, Y. Multi-behavior recommendation with graph convolutional networks. In *SIGIR* (2020).
- [20] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [21] KIPF, T. N., AND WELING, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [22] KIPF, T. N., AND WELING, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [23] KOREN, Y., BELL, R., AND VOLINSKY, C. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
- [24] LI, C., LIU, Z., WU, M., XU, Y., ZHAO, H., HUANG, P., KANG, G., CHEN, Q., LI, W., AND LEE, D. L. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management* (2019), pp. 2615–2623.
- [25] LI, F., CHEN, Z., WANG, P., REN, Y., ZHANG, D., AND ZHU, X. Graph intention network for click-through rate prediction in sponsored search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019* (2019), B. Piwowarski, M. Chevalier, É. Gaussier, Y. Maarek, J. Nie, and F. Scholer, Eds., ACM,

- pp. 961–964.
- [26] LINMEI, H., YANG, T., SHI, C., JI, H., AND LI, X. Heterogeneous graph attention networks for semi-supervised short text classification. In *EMNLP-IJCNLP* (2019), pp. 4821–4830.
 - [27] LIU, Z., LIAN, J., YANG, J., LIAN, D., AND XIE, X. 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* (2020), pp. 289–298.
 - [28] MCCALLUM, A., NIGAM, K., AND UNGAR, L. H. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000* (2000), R. Ramakrishnan, S. J. Stolfo, R. J. Bayardo, and I. Parsa, Eds., ACM, pp. 169–178.
 - [29] RENDLE, S., FREUDENTHALER, C., GANTNER, Z., AND SCHMIDT-THIEME, L. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
 - [30] SABOUR, S., FROSST, N., AND HINTON, G. E. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA* (2017), I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., pp. 3856–3866.
 - [31] SEDHAIN, S., MENON, A. K., SANNER, S., AND XIE, L. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web* (2015), pp. 111–112.
 - [32] SU, X., AND KHOSHGOFTAAR, T. M. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).
 - [33] TAN, Q., ZHANG, J., YAO, J., LIU, N., ZHOU, J., YANG, H., AND HU, X. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (2021), pp. 598–606.
 - [34] TANG, H., LIU, J., ZHAO, M., AND GONG, X. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Fourteenth ACM Conference on Recommender Systems* (2020), pp. 269–278.
 - [35] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., AND POLOSUKHIN, I. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
 - [36] VELIČKOVIĆ, P., CUCURULL, G., CASANOVA, A., ROMERO, A., LIO, P., AND BENGIO, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
 - [37] WANG, X., HE, X., CAO, Y., LIU, M., AND CHUA, T.-S. Kgat: Knowledge graph attention network for recommendation. In *SIGKDD* (2019), pp. 950–958.
 - [38] WANG, X., HE, X., WANG, M., FENG, F., AND CHUA, T.-S. Neural graph collaborative filtering. In *SIGIR* (2019), pp. 165–174.
 - [39] WANG, X., HUANG, T., WANG, D., YUAN, Y., LIU, Z., HE, X., AND CHUA, T.-S. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the Web Conference 2021* (2021), pp. 878–887.
 - [40] WANG, X., JIN, H., ZHANG, A., HE, X., XU, T., AND CHUA, T.-S. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (2020), pp. 1001–1010.
 - [41] WU, L., LIN, H., TAN, C., GAO, Z., AND LI, S. Z. Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering* (2021).
 - [42] XIA, L., HUANG, C., XU, Y., DAI, P., ZHANG, B., AND BO, L. Multiplex behavioral relation learning for recommendation via memory augmented transformer network. In *SIGIR* (2020), pp. 2397–2406.
 - [43] XIA, L., HUANG, C., XU, Y., DAI, P., ZHANG, X., YANG, H., PEI, J., AND BO, L. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 4486–4493.
 - [44] XIA, L., XU, Y., HUANG, C., DAI, P., AND BO, L. Graph meta network for multi-behavior recommendation. In *SIGIR* (2021), pp. 757–766.
 - [45] XUE, H.-J., DAI, X., ZHANG, J., HUANG, S., AND CHEN, J. Deep matrix factorization models for recommender systems. In *IJCAI* (2017), vol. 17, Melbourne, Australia, pp. 3203–3209.
 - [46] ZAHRA, S., GHAZANFAR, M. A., KHALID, A., AZAM, M. A., NAEEM, U., AND PRÜGEL-BENNETT, A. Novel centroid selection approaches for kmeans-clustering based recommender systems. *Inf. Sci.* 320 (2015), 156–189.
 - [47] ZHANG, C., SONG, D., HUANG, C., SWAMI, A., AND CHAWLA, N. V. Heterogeneous graph neural network. In *SIGKDD* (2019), pp. 793–803.
 - [48] ZHANG, J., SHI, X., ZHAO, S., AND KING, I. Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. *arXiv preprint arXiv:1905.13129* (2019).
 - [49] ZHANG, W., MAO, J., CAO, Y., AND XU, C. Multiplex graph neural networks for multi-behavior recommendation. In *CIKM* (2020), pp. 2313–2316.