Beyond Preferences: Enriching User Profiles for Effective E-commerce Recommendations

Jingyu Xu, Zhengwei Yang, *Student Member, IEEE*, Zheng Wang, *Senior Member, IEEE*

***Abstract*—Recommender systems have become a fundamental service in most E-Commerce platforms. Recently, some efforts to extract multi-interests from users' historical records have demonstrated superior performance. However, aside from historical records, the user profile contains rich semantic information for interest extraction and inherently regulates the user's interests. Existing works mainly overlook that a user's interests have: 1) group influence; 2) multi-level preference; and 3) time relevance. To this end, a novel Enhanced User Profile-based Multi-interest Model (E-UPMiM) for recommendation is proposed to integrate historical records with enhanced user profiles with social relationship information to model users' multi-interests effectively. By leveraging these multiple data sources, our model aims to provide users with more accurate and personalized recommendation. We propose to extract user preferences with three corresponding components: 1) integrated input with enhanced and updated user profile with social relationship information to meet user's grouping needs; 2) multi-interest extraction module to obtain complex multiple interest representations; and 3) time-aware ranking module to adjust the order of recommendation list dynamically. Extensive experiments on three public datasets show that E-UPMiM significantly outperforms state-of-the-art recommendation models. From both qualitative and quantitative perspectives, the results demonstrate improvements in recommendation accuracy, personalization, and robustness to changes in user preferences.**

***Index Terms*—Recommendation, User Profile, User Preference, Graph Convolutional Network, Time-aware Ranking**

I. INTRODUCTION

ONLINE E-commerce platforms, such as Amazon, eBay, and Taobao, have attracted many users to online shopping. However, the massive products presented on those platforms lead users to have trouble finding the most desired items. To address this issue, personalized recommender systems have been developed and play an increasingly crucial role in online services. As the systems are built to provide customized services to users, the human-centered experience during the whole process is critically important.

Current recommendation algorithms, such as KGAT [1] and GRU4Rec [2], mainly use historical records to extract features for generating recommendation lists, ignoring the rich semantic information inside user profiles. This leads to a poorer understanding of user interests. Recently, a novel user-aware candidate matching model, UMI [3], outperforms other state-of-the-art models by introducing user profile information in the item retrieval stage.

The user profile is a collection of user descriptions that includes static demographic information and some labels from previous behaviors [4]. With user profiles, rich semantic information can be extracted and considered when retrieving candidate recommendation lists for users. However, traditional recommender systems and many other modern algorithms fail to recognize this. To better extract user interests for recommendation, we analyze users' interests with such a powerful tool, making recommendation results more personalized and accurate.

Despite this, we notice that the user profile information used by current research tends to be static features like gender, age, OS platform, etc. We argue that these features alone cannot accurately reflect user interests. Therefore, this study introduces an enhanced and enriched user profile to promote the precision of user interest mining. This enriched profile includes the demographic features as usual, our findings from previous research, and statistical analysis of datasets.

Previous studies have demonstrated that a person's interest has three following characteristics: 1) it has grouping needs [5]; 2) it is multi-preference [3], [6]–[8]; and 3) it has time-relevance [9]–[11]. For example, a person who has a lovely pet might be interested in pet supplies when shopping and the interest in dresses is highly related to the gender of a user. Besides, a person may watch a movie he/she has never seen just because the main actor is his/her favorite star, and a man may want to continue watching other movies of the same genre after enjoying a fantastic sci-fi movie. These hidden connections are not usually indicated by his/her historical records but by his/her user profile [3]. Another example user with four watched movies is shown in Fig. 1. In this example, the user profile is provided on the left, which contains gender, age, occupation, lifestyle, etc., along with his historical records. On the right, the user's friend's historical records can also be obtained to infer the target user's interests. From the user profile (male, 26), we may infer that he might like action movies. And from his friends' interests, we may find some commonly shared interests to infer his interests. Moreover, from the movie sequence, we can see that his interests are multiple, including cartoons, action, and fantasy. Also, interest consistency can be seen in this example figure, hinting that users' interests have time relevance.

To verify the above preference analysis, a statistical analysis is performed on the Movielens and Fliggy datasets, shown later in Sec. **??**. Based on both theoretical and empirical findings, this study attempts to enhance and enrich the user

Jingyu Xu, Zhengwei Yang, and Zheng Wang are with the National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence, School of Computer Science, Wuhan University, Wuhan, 430072, China, and also with the Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan, 430072, China (e-mail: wangzwhu@whu.edu.cn). (*Jingyu Xu and Zhengwei Yang contributed equally to this work.*) (*Corresponding author: Zheng Wang.*)

Fig. 1. An example user with four watched movies. On the left is his user profile; on the right is his own watched sequence, along with his friend's watched sequence. The movie sequences of both users have been categorized by interest groups.

profile for better recommendations from three corresponding perspectives.

To begin with, it is observed that people are highly socialized and influenced by their groups due to recent advances in social networking, which make communicating among individuals with similar backgrounds (e.g., occupations, hobbies) more convenient. Previous studies on human behaviors have also revealed two basic principles that describe the influence of social relationships between users: 1) users that are socially connected tend to share similar preferences, which is referred as social homophily [12], and 2) the behavior of a user can be easily influenced by his/her friends, like co-purchasing the same item or co-rating the same movie, which is referred as social influence [13]. For example, consider the male user in Fig. 1. It shows that his friend loves action movies, which influences the target user to also give a high rating to the movie *Avatar*. Although he rates only two action movies, it can be inferred from his behaviors and friend's preferences that he might also enjoy action movies. Consequently, the social relationship information is added as a tag to the enhanced user profile for extracting user preferences from a group perspective.

Besides incorporating social relationship information, it is crucial to acknowledge that users' interests are inherently diverse and multifaceted. To better capture users' complex interests, Graph Convolution Network (GCN) and Capsule Network (CapsNet) are incorporated to form the multi-interest extraction module following the practice outlined in [6] and [3]. GCN is used to model high-order user behaviors [6], which effectively solves the problem that modeling only low-order behavior makes preference learning less precise, while CapsNet is proposed to extract multiple interests [3], [7], [8], which solves the problem that extracting user interests into a single vector is not expressive enough for personalized service. In short, these two components are utilized together to effectively model a user's multiple interests, thereby providing a more detailed and multi-grained user profile.

Moreover, users' interests have time-relevance [9]. For instance, when a girl has purchased lipstick, she is unlikely to buy the same or similar products until she is tired of or runs out of the current one. Therefore, a time-aware ranking module is proposed to meet the demand for the consistency and temporal shift in a user's short-term interests [11]. During the ranking stage, this module dynamically adjusts the order of the recommendation list, making recommendation results more precise. Adding this module can significantly improve performance on order-sensitive recommendation scenarios. Additionally, this module can determine whether a user's interest shift is temporally consistent, enriching the user profile from a time-related perspective.

In summary, our analysis of user behaviors reveals three key characteristics of user interests: group influence, multi-level preference, and time relevance. Accordingly, a novel **E**nhanced **U**ser-**P**rofile-based **M**ulti-**i**nterest **M**odel (E-UPMiM) is proposed for recommendation. The E-UPMiM consists of three main components to relate the above three characteristics accordingly: integrated input with enhanced user profile containing social relationship information, multi-grained multi-interest extraction module, and time-aware ranking module. As a result, the user's profile is continuously enriched and updated at the end of each training epoch, ensuring that the recommendations remain accurate and personalized.

The contributions can be summarized threefold:

- We conduct a comprehensive analysis of user behaviors, identifying three key characteristics of user interests: group influence, multi-level preference, and time relevance.
- We propose a novel **E**nhanced **U**ser-**P**rofile-based **M**ulti-**i**nterest **M**odel (E-UPMiM) for recommendation with three components to relate the above three characteristics accordingly.
- We validate the effectiveness of E-UPMiM through extensive experiments on the three real-world datasets, demonstrating significant improvements in recommendation accuracy and personalization.

II. Related Work

A. User Profile and Relationship Modeling

User profile modeling is an important task in terms of personalized recommendation services. Traditionally, the user profile research adopts a surveying-grouping-tagging method, which heavily relies on manual design as the features are often devised in advance. However, as deep learning techniques develop [14], more deep learning-based methods are proposed, such as an adversarial substructure representation learning framework [15], YouTubeDNN [16], etc. User profiles can help understand users, provide customized products and services, improve user satisfaction, and, thus, have widely been applied in various applications, including recommender systems [3], [17]. Nevertheless, existing methods mainly observe user interests from his or her behaviors only and overlook that users' interests are determined not only by historical records but by the user profile itself. Recently, a novel user-aware candidate matching model, UMI [3], is proposed to leverage user profile information in the item retrieval stage, which outperforms other state-of-the-art models, opening up a new path to recommendation scenes.

Relationship modeling is another hot topic in the recommendation area. It is based on the idea that socially connected users tend to share similar preferences and that the behaviors of a user can be easily influenced by his/her friends [12], [13], [18]. To characterize the above social relationship impact on user preferences, many social recommendation methods have been proposed, like enhancing user preference modeling motivated by the social principle of homophily [5], focusing on influence from trust relations [19], and exploiting different relations with strong or weak ties [20]. Recent progress of Graph Neural Network (GNN) has further boosted the development of graph-based social recommendation [21], as the propagation mechanism in GNN is well suited for modeling influence spread process in social recommendation scenes [22]–[25].

B. Graph Convolution Network

GCN is an important tool in many areas, including recommendation [21], [26]. Specifically, GCNs adopt embedding propagation to aggregate neighborhood embedding iteratively. Stacking the propagation layers allows each node to access high-order neighbors' information rather than only the first-order neighbors', as traditional methods do. With its advantages in handling structural data and exploring structural information, GCN-based methods have become the new state-of-the-art approaches in recommender systems. KGAT [1] utilizes Knowledge Graph (KG) with GCN and attention mechanism to enrich the user-item bipartite graph with knowledge and increase explainability. LightGCN [27] examines the structure of GCN to reduce redundant components and achieves greater performance than other GCN models. Similarly, we adopt GCN to model high-order user interests, but unlike other traditional GCN models, we model multi-level interest in a multi-interest way via CapsNet.

C. Multi-Interest Extraction

To capture a user's multiple interests in the retrieval stage, the capsule network [28] (shortly as CapsNet) has been widely used to generate multiple interest vectors in recommender systems [7], [29]. Furthermore, ComiRec [8] integrates a controllable balance value function with a greedy inference algorithm for item aggregation after retrieving items with CapsNet to balance accuracy and diversity. The sparse interest network [30] generates multi-interest embeddings from a large set of intention prototypes. Another deep multi-interest network [31] is proposed to model users' multi-interest for the click-through rate (CTR) prediction task in E-commerce platforms [32]. However, these methods only concatenate static user profiles after extracting multi-interests, which cannot fully use user profile information because various, instead of static user profiles, are important to incorporate for diverse interest learning. Differently, our proposed method integrates user profiles and social information with historical records to better capture user interests. Besides, we design a user profile-based interest refining module to use the user profile fully.

D. Time-aware Recommendation

Recent studies [10], [33]–[37] have focused on leveraging the time intervals among users' behaviors to better capture users' preferences, for which traditional sequential architectures are insufficient. Mei and Eisner [33] propose a neural model that allows past events to influence future predictions in a complex and realistic fashion. Zhao et al. [34] introduce a distance gate based on a recurrent network to control the short-term and long-term point-of-interest updates. TMI-GNN [35] yields refined intention representations by injecting two-level temporal information, achieving state-of-the-art performance in the Session-Based Recommendation (SBR) task. PTGCN [36] models the temporal dynamics between user-item interactions by defining a time-aware graph convolution operation and learning the dynamic representations of users and items simultaneously on the bipartite graph with a self-attention aggregator. CDTR [10] proposes a causally debiased time-aware recommender framework to learn user preference accurately. CoNCARS [37] leverages a deep Convolutional Neural Network (CNN) to learn the nonlinear user-item-time correlations.

We argue that these methods will bring noise and increase computational cost and time complexity. Differently, we conduct preliminary time processing in CapsNet and deal with time factors in the ranking stage, which reduces training costs and brings no noise.

III. METHOD

In this section, we present the proposed E-UPMiM model in detail. As illustrated in Fig. 2, the proposed E-UPMiM consists of three main components: integrated input with enhanced user profile, multi-interest extraction module, and time-aware ranking module. In the following, we first present some statistics-based findings and the formal problem setting. Then, we describe these components, followed by the prediction and model optimization process.

A. Statistics-based Findings

Before formulating the problem, we want to present our statistical analysis and findings. We define two users as friends if they have the same behavior (in this case, rating) on the

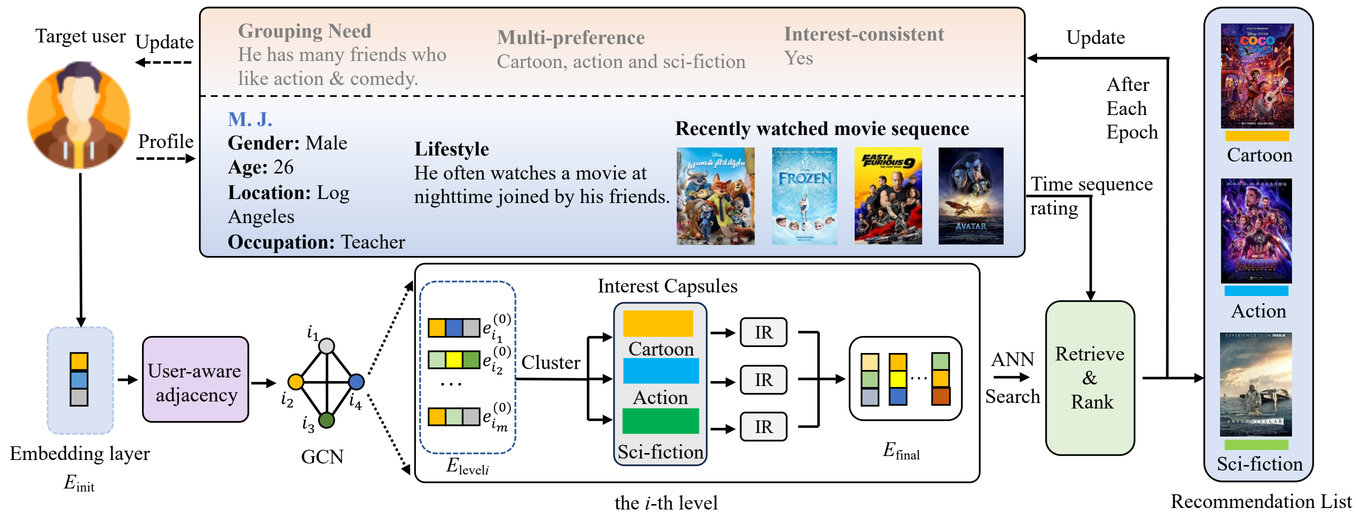

Fig. 2. Framework of the proposed method. The user profile on the upper side includes static attributes and watched history in sequence. After each epoch, the profile will be updated and fed into the model input starting from epoch 2. The IR module is displayed in Fig. 3; and the time-aware ranking module is displayed in detail in Fig. 4.

TABLE I
THE STATISTICS OF USERS WITH SPECIFIC NUMBER OF FRIENDS ON THE MOVIELENS AND THE FLIGGY DATASET.

Dataset	#Users with certain number of friends					
	1	2	3	4	5	over 5
Movielens	278	2,040	2,625	403	375	319
Fliggy	25,532	30,287	27,877	43,991	37185	131,249

TABLE II
THE STATISTICS OF FRIEND PAIRS AND AVERAGE NUMBER OF SIMULTANEOUSLY-INTERACTED ITEMS IN EACH FRIEND PAIR (SHORT AS AVERAGE ITEMS) ON THE MOVIELENS AND THE FLIGGY DATASET.

Dataset	#friend pair	#average items
Movielens	13,415	6
Fliggy	1,218,220	102

same item simultaneously or around the same timestamp [5]. Table I shows the number of users with a specific number of friends on the Movielens and Fliggy datasets, respectively. According to our definition, the table shows that most users in the Movielens dataset have two or three friends and that most users in the Fliggy dataset have more than five friends. Further, we report how many items, on average, are interacted with simultaneously between each friend pair in Table II. This finding suggests that users' preferences might be naturally influenced by his/her friends, just as the motivation figure shows.

In addition, table III shows how many categories a user may have interests in. From the table, it can be drawn that most users might be interested in at least three categories in the movie genres.

Finally, we analyze the temporal interest shift in the Movielens dataset and find that nearly 70% of the users have certain interest consistency.

With the above statistical analysis in hand, we can proceed to our problem formulation and method overview.

TABLE III
THE STATISTICS OF USERS WITH SPECIFIC NUMBER OF INTEREST CATEGORIES ON THE MOVIELENS DATASET.

#Users with certain number of interest categories				
3	4	5	6	over 6
307	412	1,059	1,239	3,023

B. Problem Formulation and Method Overview

1) Problem Formulation: Let $\boldsymbol{U}$ denote a set of users, $\boldsymbol{I}$ denote a set of items (known as item pool). For a given user u, Iu is his/her interacted items (x_1, x_2, x_3, ..., x_t), where x_i is the ith item interacted by the user and t is the historical behavior length. P_u is his/her user profile. The ultimate goal is to recommend what the user needs. However, modern recommender systems usually adopt a two-stage approach, the retrieval stage and the ranking stage. The retrieval stage retrieves top-N candidate items from the item pool while the ranking stage is designed to rank the items based on specific criteria. Our paper mainly focuses on the retrieval stage by improving the modeling of user embeddings. The task for the retrieval stage in recommender systems is to retrieve a subset of items related to the user interests from $\boldsymbol{I}$. Generally, the task in the retrieval stage with multi-interest extraction can be formulated as follows:

$$relevance_{u,t} = \max_{1 \le k \le K} ((i_u^k)^T e_t) \tag{1}$$

where i_u^k denotes kth interest vector for the user u and e_t denotes the representation vector for target item x_t. $relevance_{u,t}$ is the relevance score between the users' interest vectors and the target item. In the latter part of the retrieval stage, we adopt an ANN approach to find the item representations closest to the user interest embeddings as the candidate items. In the ranking stage, we introduce a time-aware ranking strategy that considers the fact that users' interests have temporal relevance.

2) Method Overview: An overview of our method is shown in Fig. 2. As mentioned above, three kinds of inputs will be fed into the model: enhanced user profile with his/her social relationship information, users' historical records, and the target item.

The original user profile mainly contains users' static attributes, such as gender, age, and occupation in the Movielens dataset. For social relationship information, we follow paper [5] to identify users who most likely share similar interests with the target user and then concatenate these users' information in the form of a tag with the target user's profile to enhance it. The user profile obtained in this way is called the initial user profile, which will be enhanced later during the training process. Users' historical records contain a list of items the users have interacted with and share the same data format with the target item, including ItemID, timestamp, etc. An embedding layer is first conducted to project these features into fixed-size dense embeddings. After this, features within a field are concatenated into a field representation. We then form the initial user profile and item embeddings by concatenating the embeddings of the corresponding fields together.

C. Multi-interest Extraction

User-aware Graph Convolution. Given the users' historical records that include users' interacted item sequences, we first transform these items into a fully connected graph G_i by taking each item e_i as a node. We then introduce $A \in R^{m\times m}$ to denote the adjacency matrix of the fully connected graph G_i, where $A_{i,j}$ indicates the relatedness between the item e_i and the item e_j. Note that we have the user profile already, so we choose to learn the adjacency matrix in a user-aware manner by introducing e_u as follows:

$$A_{i,j} = \text{sigmoid}((e_i \otimes e_j) \cdot e_u) \tag{2}$$

where $\otimes$ is Hadamard product, $\cdot$ is inner product and $sigmoid$ is the activation function. We can see that we utilize the initial user profile information in the retrieval stage, making the candidate-matching process more personalized.

We perform graph convolution operation over G_i as follows following the common practices [6]:

$$I^{(1)} = [e_1, e_2, ..., e_m], \tag{3}$$

$$I^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}I^l W) \tag{4}$$

$$\tilde{D}^{-\frac{1}{2}} = I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \tag{5}$$

where I^l denotes the item representations in the lth layer ($l \in 1, 2, ..., L$), $\sigma(\cdot)$ denotes the LeakyReLU, D denotes degree matrix of A, I denotes the identity matrix, and W denotes the trainable parameter. The parameter matrix W is shared for all L layers. Sharing the parameter matrix W reduces the model complexity while facilitating feature aggregation from the high-order neighbors. The item representations obtained in each layer can reflect the user's multi-level preferences, leading to more personalized interest learning.

Multi-Interest Routing. For each level i in the previous GCN, the items $I^i = [e^i_1, e^i_2, ..., e^i_m]$ are fed into a capsule network to generate multi-interest vectors. In Capsule Network, a capsule is a cluster where input vectors related to the same category are aggregated together. Given the the item representations $I^i = [e^i_1, e^i_2, ..., e^i_m]$ at level i, the jth capsule calculates the interest vector h_j as follows:

$$h_j = \sum_{i=1}^{m} c_{ij} W_j e^i, \tag{6}$$

where W_j is the transformation matrix, and c_{ij} are the coupling coefficients, indicating the clustering probability for item e^i under the jth interest capsule. Following the dynamic routing mechanism proposed in CapsNet [3], [8], we have equations as follows:

$$c_{ij} = \frac{\exp b_{ij}}{\sum_{k=1}^{K} \exp b_{ik}}, \tag{7}$$

where b_{ij} are the initial logits, defined as follows:

$$b_{ij} = (W_j e_i)^T s_j, \tag{8}$$

where s_j is the "squashed" interest vector:

$$s_j = squash(h_j) = \frac{||h_j||^2}{||h_j||^2 + 1} \frac{h_j}{||h_j||^2} \tag{9}$$

where h_j is the total input of capsule j. From equation 6 to equation 9, we can see that the calculation of the output s_j relies on itself. As a result, a dynamic routing process is designed to update its value iteratively. We first initialize b_{ij} with Gaussian distribution.

Given the above process, we have $L \cdot K$ interest capsules after iteratively updating values.

These interest capsules cluster users' interests iteratively, guiding us to explore the user's interest categories and enhance the user profile with such information.

User-profile-based interest relearning. With $L \cdot K$ interest capsules on hand, we further leverage the user profile information to enhance the user interest learning process following the practice of the UMI [3], as shown in Fig. 3. Firstly, we define the relevance weights for the user profile as follows:

$$w_k = \text{sigmoid}(\text{ReLu}(o^i_k, e^m_u)) \tag{10}$$

where o_k is a capsule network on level i, and e^m_u is the mth field of user profile.

Based on the weights w_k, the user profile embedding can be formulated as:

$$e^k_u = \text{Concat}(w_{k1}(e^1_u), w_{k2}(e^2_u), ..., w_{km}(e^m_u)) \tag{11}$$

where w_{km} is the mth element of w_k. After this, we derive the final kth user interest as follows:

$$output^k_u = \text{MLP}([e^k_u, o^i_k]) \tag{12}$$

where $output^k_u$ is the user interest embedding for a given user and can be used later. When serving, we use Faiss (or ANN) to randomly sample retrieved items and then calculate the relevance score according to equation (1).

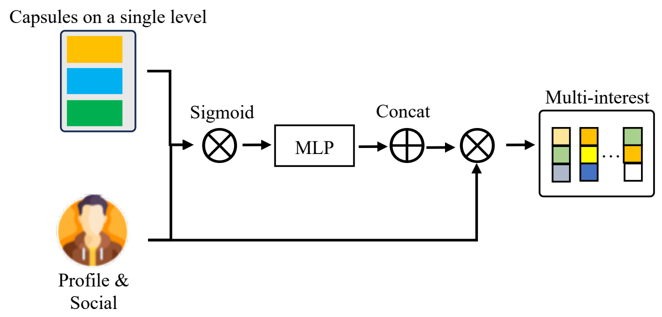

Fig. 3. Details of Interest Relearning (IR) module. The user profile information is further utilized to enhance the interest learning process.

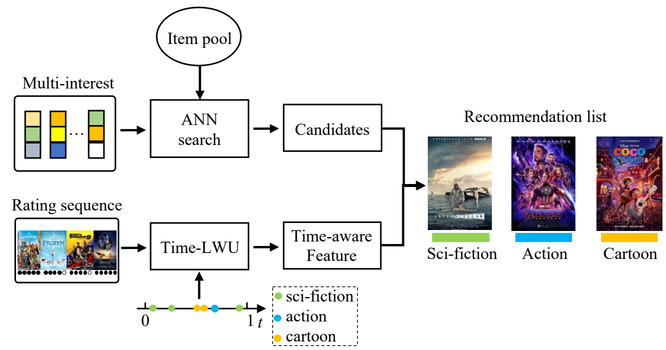

Fig. 4. Details of time-aware ranking module. In the module, we use time factors to get the time-aware feature, project these features into the candidates' scores, and rank them.

D. Time-aware Ranking and Training Process

Time-aware Ranking. Before giving top-N recommended results, the retrieved items will be ranked. Current methods just simply order them based on the relevance scores above. However, we argue that users' interests might have temporal consistency or decay over time. So, in addition to pair-wise score-based rank, we adopt a time-aware rank method, as shown in Fig. 4. Firstly, we use a sequence to include the target user's interacted items' interaction time and rating score, forming a collective time vector E_{ct}. Later, we generate time-aware features E_{time} in the Latest Time Weight Unit (namely Time-LWU) based on categories' appearance frequency, how recently they appear, mean rating score, user's interest-consistency feature, etc. Specifically, for how recently each category appears, the Time-LWU will add weights with an LWU strategy, meaning the most recent category will receive the greatest weight value. It should be noted that we consider the user's interest-consistency feature, which indicates whether the user may click on the same category item or not. Also, we consider the frequency of each category to avoid popularity bias to some extent. Finally, we rank the candidate items with consideration of the above time-aware features. In this way, we consider not only the relevance score but also the interest consistency factor.

Also, we add a label to the user profile indicating whether the user's temporal interest is consistent in the module to enhance the user's profile in a time-sensitive fashion.

Training Process. After each training epoch, the user profile will be enhanced; we then update the user profile of each user at the end of the training process in the next epoch, which means the input of the user profile comes from the output of the last iteration starting at epoch 2 (for epoch 1, the initial user profile usually comes from the dataset directly).

E. Prediction and Model Optimization

1) Prediction: We calculate the recommendation score using an inner product between the user interest vectors on level i and the target item vector:

$$rec_{u,t}^{i} = e_u^i e^t \tag{13}$$

where $rec_{u,t}^{i}$ denotes the recommendation score between the user and the target item on level i. Note that we have L level(s) in the GCN operation; we obtain the final recommendation score across different layers with a max-pooling operation:

$$rec_{u,t} = \max(rec_{u,t}^{1}, rec_{u,t}^{2}, ..., rec_{u,t}^{L}) \tag{14}$$

2) Model Optimization: Different from the loss functions currently used in multi-interest models like ComiRec [8] and UMI [3], we choose to define losses for each level [6] and formulate the final loss as follows:

$$\mathcal{L}_{all} = \sum_{l=0}^{L} \mathcal{L}_1 + \theta \mathcal{L}_2 \tag{15}$$

$$\mathcal{L}_1 = -\sum_{u,t} \log rec_{u,t}^{i} \tag{16}$$

where $rec_{u,t}^{i}$ is the likelihood that user u will interact with item e_t, $\mathcal{L}_2$ is the all model parameters, and θ denotes the hyperparameter.

IV. Experiments

In this section, we first compare the proposed E-UPMiM with the current state-of-the-art solutions on three open datasets and perform a series of ablation studies to verify the enhanced user profile, multi-interest extraction, and time-aware ranking module, respectively. Then, several case studies are conducted to demonstrate the effectiveness of E-UPMiM intuitively. Finally, we further analyze the impact of parameter settings on our proposed model.

A. Experimental Settings

1) Datasets: We conduct experiments on three open real-world datasets.

The first one is Movielens. Movielens is a widely used and acknowledged dataset containing user ratings (integers ranging from 1 to 5) on movies. Here, we choose its 1M version. To reduce noise, we filter out users with less than 5 interactions and treat ratings equal to or over 3 as positive and less than 3 as negative samples.

The second dataset is the Fliggy dataset [38]. The Fliggy dataset contains all behaviors (including clicks, favorites, adding, and purchases) of approximately 2 million random

TABLE IV
STATISTICS OF DATASETS AFTER PREPROCESSING.

Dataset	#Users	#Items	#Interactions
Movielens	6,040	3,416	999,611
Fliggy	277,662	39,784	2,370,933
Toys & Games	313,557	241,657	6,212,901

users from June 3, 2019, to June 3, 2021, on the travel platform Fliggy. Here, we only use the purchase behavior and follow the same preprocess procedure as we do on the Movielens dataset.

Another dataset is the Toys & Games sub-dataset from the Amazon Product Data [39]. It contains users' ratings and some product info. Here, due to its lack of user profile, the initial user profile embedding is initialized with Gaussian distribution and updated with our model after epoch 1.

Table IV summarizes detailed statistics of the three datasets after preprocessing.

2) Baselines: We compare the proposed E-UPMiM with the following state-of-the-art recommendation methods:

YouTubeDNN [16]: YouTubeDNN adopts a mean-pooling operation over the embeddings of the user historical items, after which the user profile embedding is concatenated. Then, an MLP layer is utilized to derive the final user representation.

GRU4Rec [2]: It utilizes the gated recurrent unit to model the session sequence for recommendation.

MIND [7]: This novel multi-interest recommendation model utilizes the capsule routing mechanism to extract multi-interest vectors.

ComiRec [8]: This is a novel multi-interest recommendation model. We use the ComiRec-DR version based on a dynamic routing mechanism.

UMI [3]: This is a state-of-the-art user-aware multi-interest learning model that fuses user profiles and a capsule network to generate recommendation lists.

All these baselines can be divided into three categories: (1) user-aware models that utilize user profile information (i.e., YouTubeDNN); (2) traditional sequential models that utilize RNN and attention mechanism (i.e., GRU4Rec); (3) multi-interest models that derive various user interest (i.e., MIMD, ComiRec, and UMI).

3) Data Preparations: For data preparations, we follow common practices to divide these three datasets by 8:1:1 as train, test, and validation, respectively. For dataset splitting, we first adopt a temporal ordering strategy to split interacted items on a sampled user because our task is time-sensitive. Then, we use ratio-based splitting following common practice instead of the leave-one-out method.

4) Hyperparameter Settings: For a fair comparison, all methods are implemented with TensorFlow 1.14 in Python 3.6 and learned with Adam optimizer [40]. We conduct our experiments on a single Linux server with 12 Intel(R) Xeon(R) CPU Platinum 8255C @2.50GHz, 64GB RAM, and 1 NVIDIA GeForce RTX 3090Ti(24GB). The learning rate and mini-batch size are set to 1e - 3 and 128 for the Movielens dataset and 1e - 3 and 256 for the Fliggy and Toys & Games datasets. The number of negative samples is 5 in the training stage for all three datasets. The number of iterations for the dynamic routing method is set to 3. The number of interest embeddings (K) for multi-interest modules is set to 3 and 4 for the Movielens and the other two datasets, respectively. For GCN in our model, we set layer number (L) to be 4 by default. We tuned the parameters of all methods over the validation and set the embedding size as 32 and 64 for the Movielens and the other two datasets, respectively.

5) Evaluation Metrics: For evaluation, we use **HitRate@N** and **NDCG@N** metrics, which have been widely used for recommendation:

$$\text{HitRate@}N = \frac{1}{|\mathbf{U}|}\sum_{u\in\mathbf{U}}\delta(|I_u \cap I| > 0) \tag{17}$$

where $\delta(\cdot)$ is the indicator function, I_u is the set of top-N recommended items for user u, and I is the set of testing items.

$$\text{NDCG@}N = \frac{\text{DCG@}N}{\text{IDCG@}N} \tag{18}$$

where $\text{DCG@}N$ is the discounted cumulative gain that considers the positions of correct recommended items and $\text{IDCG@}N$ is the ideal discounted cumulative gain or the maximum possible value of $\text{DCG@}N$.

B. Overall Performance

Table V summarizes the overall performance of all models. Here, we make the following observations.

First, the user-aware model (YouTubeDNN) achieves relatively higher scores in candidate matching than MIND and Comi-Rec, suggesting that user profile information is of great importance in user interest learning. Another explanation for this is that MIND and Comi-Rec directly use the user representation selected by the target item for those negatives, which makes the negatives too "easy" to recognize during the model training.

Second, traditional sequential models (GRU4Rec) that utilize RNN and the attention mechanism are difficult to perform better. Compared with multi-interest models, they are unsuitable for complex and various user interest modeling. This is because traditional sequential models mainly adopt a single vector approach to represent user embedding and item embedding, which, as discussed earlier, is not comprehensive enough to extract the user's preference.

Third, multi-interest models (MIND, Comi-Rec, UMI) perform much better than traditional sequential models. This suggests that by utilizing multiple interest vectors and the capsule network for dynamic interest routing, multi-interest models could achieve a more precise preference understanding than models that use only a single vector.

Specifically, UMI performs better than the other multi-interest models above. UMI adopts a user-aware candidate-matching method by utilizing user profile information during the candidate retrieval stage. This result and YouTubeDNN's performance show that user profiles can produce a great performance boost in personalized recommendation.

Our proposed method outperforms state-of-the-art baseline models in most settings, showing strong effectiveness in

TABLE V
THE OVERALL PERFORMANCE OF DIFFERENT METHODS ON THE THREE DATASETS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE. THE RESULTS ARE PERCENTAGE NUMBERS WITH "%" OMITTED.

	Movielens				Fliggy				Toys & Games			
	metrics@10		metrics@50		metrics@10		metrics@50		metrics@10		metrics@50	
	HitRate	NDCG	HitRate	NDCG	HitRate	NDCG	HitRate	NDCG	HitRate	NDCG	HitRate	NDCG
YouTubeDNN	7.16	9.59	26.69	8.24	10.83	6.90	8.24	6.94	11.27	10.35	22.79	15.35
GRU4Rec	6.84	8.96	20.44	8.45	9.25	4.99	7.26	5.18	9.79	8.94	21.45	13.23
MIND	6.21	9.22	22.42	9.08	10.97	7.45	8.34	7.51	10.23	9.67	22.32	14.87
ComiRec	6.12	9.78	24.12	9.13	10.93	6.56	8.37	6.62	11.44	10.27	22.67	15.02
UMI	7.51	10.32	27.81	9.86	11.22	10.67	11.35	**9.29**	14.47	12.66	25.91	18.89
Ours	**7.79**	**11.76**	**33.61**	**12.23**	**12.08**	**11.94**	**13.96**	6.78	**16.94**	**13.78**	**26.49**	**20.78**

recommendation accuracy. 1) Our E-UPMiM surpasses other multi-interest models like MIND [7] and ComiRec [8], attributed to the fact that we utilize the user profile information, which is ignored in the above methods. 2) Our E-UPMiM outperforms traditional models like YouTubeDNN [16] and GRU4Rec [2] by a large margin. This implies that by utilizing multiple interest vectors and the capsule network for dynamic interest routing, our model could achieve a more precise preference understanding than models that use only a single vector. 3) Specifically, regarding the NDCG metric, our E-UPMiM method outperforms the UMI [3] model in most experiment settings, demonstrating that our time-aware ranking module shows potential in order-related recommendation scenes. Further, our E-UPMiM model surpasses the UMI model in all settings regarding the HitRate metric, showing the value of an enhanced user profile and multi-level design.

We also notice that our model does not work well regarding NDCG@50 on the Fliggy dataset, and we make the following analysis. The Fliggy dataset contains many yearly factors, while our time-aware ranking module mainly captures the user's temporal interest shift. When the item count is 10, the temporal factor takes the lead, but when the item count reaches 50, the yearly factors play out, leading to inferior performance on NDCG@50.

C. Ablation Study

We conduct an ablation study for each design choice in E-UPMiM to validate their effectiveness. Specifically, these factors include enhanced user profile (Profile), multi-interest extraction (M-L), and time-aware ranking (T-A). Table VI reports the performance of these variants and the full E-UPMiM model on the Movielens dataset. Here, we can make the following observations.

Firstly, we add social relationship information to the baseline and the result shows a slight performance boost, which verifies our first idea that users are socialized and have grouping needs. Secondly, we add a multi-interest extraction module (mainly GCN) to the baseline model and find a large performance jump over the baseline model, showing a strong ability to retrieve users' multi-interests by learning higher-order user interests and exploiting more information from rich semantic interactions. Also, we can find that adding social information alone is superior to adding the multi-interest module, which demonstrates the effectiveness of multi-interest

TABLE VI
THE ABLATION STUDY OF E-UPMIM ON MOVIELENS DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE.

Module			Movielens	
Profile	M-L	T-A	HR@10	NDCG@10
○	○	○	7.635	11.242
●	○	○	7.643	11.269
○	●	○	7.691	11.291
○	○	●	7.637	11.280
●	●	○	7.785	11.345
●	○	●	7.650	11.366
○	●	●	7.695	11.417
●	●	●	**7.790**	**11.456**

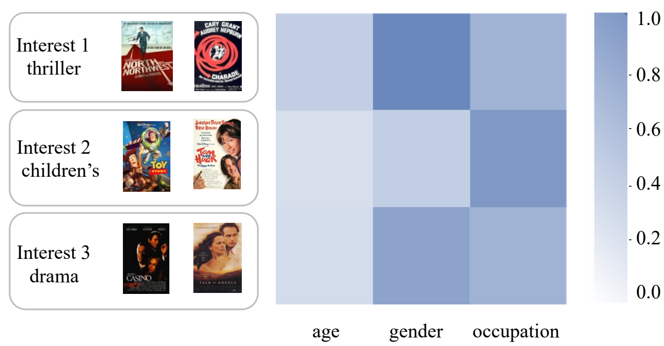

Fig. 5. UserA's user profile and its impact on the user's preferences (Best viewed in color).

learning. Thirdly, we add a time-aware ranking module to the baseline model to see if performance on the NDCG metric can be improved, and the result is within our expectations. In this experiment setting, although the HR metric doesn't increase, the NDCG metric sees some improvements. This is reasonable since this module ranks all candidate items considering the time-related factor. As we explained earlier, the NDCG metric evaluates recommendation results by weighing positions and comparing them with true/ideal conditions, leading to higher scores with more accurate relevance order.

Later, we perform further experiments to see these modules' performances under different combinations. The first one is user profile and multi-interest learning. We can see that this experiment setting has a close performance with the proposed model on HR metrics and improves much compared with settings with the user profile module alone or with the multi-interest module alone. The second one is user profile and time-aware ranking. It can be seen that there is a slight performance gain compared with settings with the user profile module alone or with the time-aware ranking module alone. And the last one is multi-interest learning and time-aware ranking. We can see that this experiment setting has pretty good performance on the NDCG metric compared with adding a multi-interest learning alone setting.

D. Case Study

Study on the impact of user profile. In this part, we randomly select a user instance (namely userA) from the Movielens dataset and perform a case study to display our model performance intuitively. UserA's historical records can be roughly categorized into six groups: drama, comedy, action, children's, thriller, and adventure. We also explore over five users who share the same interests from the dataset and mine their interests. Not surprisingly, we find relevant interests from his friends, which gives feedback to our interest extraction on userA. Fig. 5 shows the user profile and its impact on the user's preference. We only show three of his six categories because these three categories we choose are typical enough to see the trend. From the figure, we can see that gender has the biggest impact on the thriller category. This figure also explains why the user likes children's movies – he is an educator.

Study on the impact of multi-interest extraction. Later, we perform graph convolution and multi-interest routing experiments, and the results are shown in Fig. 6 and Fig. 7. Fig. 6 shows the interest level distribution of 50 randomly sampled users from the Movielens dataset, where darker colors indicate more impact on recommendation generation. From Fig. 6, we can see a trend that these users' lower levels have less impact than higher levels, which indicates that these users' preferences are rather multiple. Fig. 7 shows the multi-interest extraction of userA (the same user in the previous part). From Fig. 7, we can see that his interests are partly divided into three categories: comedy, adventure, and drama. Each category can be used to generate a set of candidate items, as shown in the right part of Fig. 7. In this manner, we can thoroughly understand users' preferences.

Study on the impact of time-aware ranking. Also, we conduct a case experiment with time-aware ranking. Before the ranking module, the recommendation lists in order are 3000 comedy/sci-fiction, 19 comedy, and 1404 drama/crime, while after the ranking module, the lists become 1404 drama/crime, 3000 comedy/sci-fiction, and 19 comedy. Compared with recommendation lists without time-aware ranking, the lists generated by the ranking module reflect the consistency of userA's short-term interests. Based on the userA's historical rating data, current methods tend to focus on comedy movies that gain higher ratings.

E. Further Analysis

In this subsection, we further investigate the impact of important parameter settings on the performance of our model.

Impact of *L* Value. Note that we stack L layers of graph convolution in E-UPMiM to reflect the user's diverse preferences in a multi-grained manner. In Fig. 8, we plot the performance changes of varying L values for the Movielens dataset. The information propagation process in the GCN allows features to pass from neighboring nodes to the target node, which enables the model to handle high-order relations. This principle can well explain that as the number of layers (L) increases from 1 to 3, both the HitRate@10 and NDCG@10 increase steadily. However, when L becomes too large, noisy information would be included to deliver adverse impact, which results in performance degradation when L continues increasing to 4. The above observation indicates that $L = 3$ is the optimal setting for the L value in the Movielens dataset.

Impact of *K* Value. To better model users' diverse interests, we use a CapsNet to generate a set of vectors (the number is controlled by K), each representing a sub-interest. So, the number of interests K controls the diversity of user preferences. Fig. 9 plots the performance changes of varying K values for the Movielens dataset. We can observe that a single interest (i.e., $K = 1$) achieves the worst performance across the two metrics, which is the common practice in most current methods. As K grows before reaching its optimal setting (where $K = 3$), the local features play a crucial role in the retrieval process.

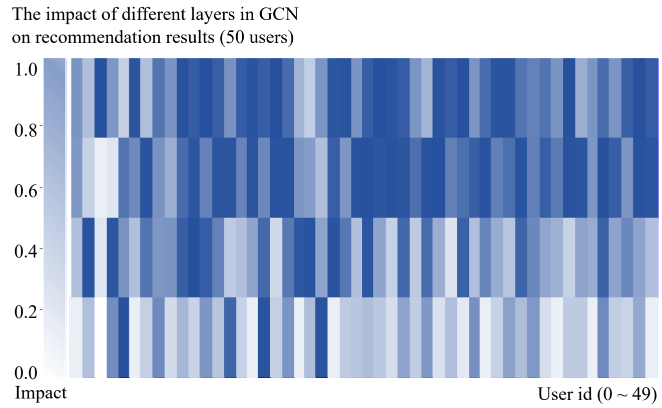

Fig. 6. Interest level distribution of 50 randomly sampled users on the Movielens dataset. Each column contains the impact of different layers in GCN on recommendation results for one user (Best viewed in color).

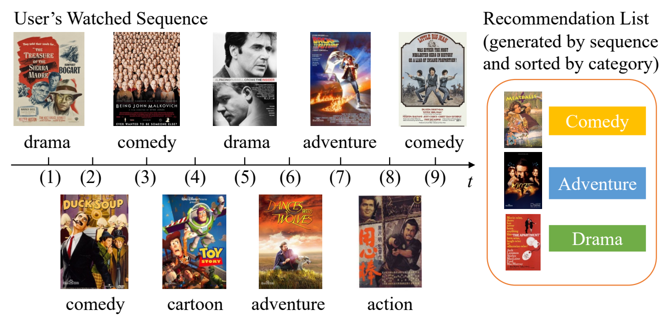

Fig. 7. Visualization of multi-interest extraction on userA. On the left is the user's watched sequence in time order; on the right is the recommendation list generated by our proposed multi-interest candidate generation model.

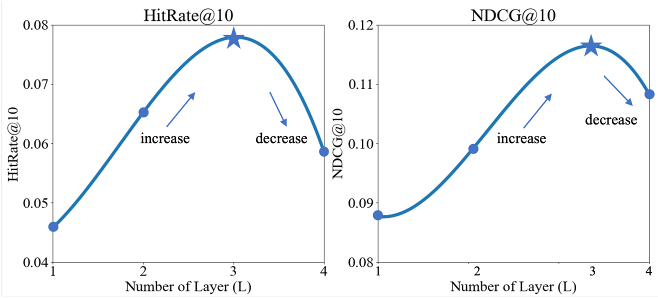

Fig. 8. The performance of different L values on the Movielens dataset.

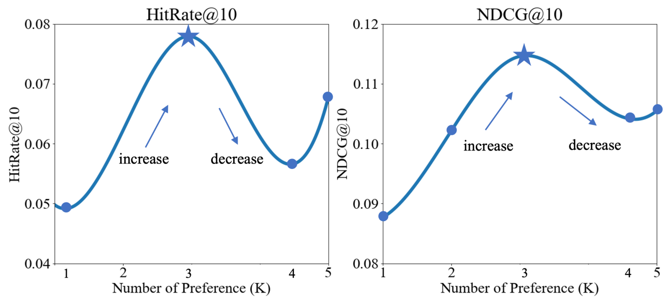

Fig. 9. The performance of different K values on the Movielens dataset.

V. Conclusion

In this paper, a novel enhanced user-profile-based multi-interest model is proposed for recommendation, enabling more precise and diverse candidate retrieving in recommenders. We introduce enhanced user profiles with social relationship information into the model as the user's preference is also regulated by profile information and influenced by social relations. We also propose Graph Convolution and Capsule Network to better exploit users' dynamic and complex interests in a multi-grained manner. Specifically, we introduce the time-aware ranking module on the basis that a user's short-term interest may be consistent. To demonstrate the effectiveness of E-UPMiM, we conducted comprehensive experiments on three real-world datasets of different sizes and compared E-UPMiM with a few baselines. Experimental results indicate that E-UPMiM significantly outperforms state-of-the-art models and improves recommendation accuracy and personalization from both qualitative and quantitative perspectives.

Although social relationship modeling and time-aware ranking have been proven effective, they are currently relatively simple. In the future, we will continue to explore more methods for these fields to achieve better recommendation results and bring convenience to countless users.

References

[1] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, New York, NY, USA: Association for Computing Machinery, 2019, 950–958. DOI: 10.1145/3292500.3330989. [Online]. Available: https://doi.org/10.1145/3292500.3330989.

[2] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, *Session-based recommendations with recurrent neural networks*, Mar. 2016. [Online]. Available: https://arxiv.org/abs/1511.06939.

[3] Z. Chai, Z. Chen, C. Li, *et al.*, "User-aware multi-interest learning for candidate matching in recommenders," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '22, New York, NY, USA: Association for Computing Machinery, 2022, 1326–1335. DOI: 10.1145/3477495.3532073. [Online]. Available: https://doi.org/10.1145/3477495.3532073.

[4] M. Diao, Z. Zhang, S. Su, S. Gao, and H. Cao, "Upon: User profile transferring across networks," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20, New York, NY, USA: Association for Computing Machinery, 2020, 265–274. DOI: 10.1145/3340531.3411964. [Online]. Available: https://doi.org/10.1145/3340531.3411964.

[5] L. Guo, H. Yin, T. Chen, X. Zhang, and K. Zheng, "Hierarchical hyperedge embedding-based representation learning for group recommendation," *ACM Transactions on Information Systems*, vol. 40, no. 1, Sep. 2021. DOI: 10.1145/3457949. [Online]. Available: https://doi.org/10.1145/3457949.

[6] Y. Tian, J. Chang, Y. Niu, Y. Song, and C. Li, "When multi-level meets multi-interest: A multi-grained neural model for sequential recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '22, New York, NY, USA: Association for Computing Machinery, 2022, 1632–1641. DOI: 10.1145/3477495.3532081. [Online]. Available: https://doi.org/10.1145/3477495.3532081.

[7] C. Li, Z. Liu, M. Wu, *et al.*, "Multi-interest network with dynamic routing for recommendation at tmall," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM '19, New York, NY, USA: Association for Computing Machinery, 2019, 2615–2623. DOI: 10.1145/3357384.3357814. [Online]. Available: https://doi.org/10.1145/3357384.3357814.

[8] Y. Cen, J. Zhang, X. Zou, C. Zhou, H. Yang, and J. Tang, "Controllable multi-interest framework for recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '20, New York, NY, USA: Association for Computing Machinery, 2020, 2942–2951. DOI: 10.1145/3394486.3403344. [Online]. Available: https://doi.org/10.1145/3394486.3403344.

[9] H. Miao, Q. Zhou, H. Song, and F. Wan, "Time interest network for click-through rate prediction," in *Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering*, ser. EITCE '22, New York, NY, USA: Association for Computing Machinery, 2023, 1159–1165.

DOI: 10.1145/3573428.3573636. [Online]. Available: https://doi.org/10.1145/3573428.3573636.

[10] L. Wang, C. Ma, X. Wu, Z. Qiu, Y. Zheng, and X. Chen, "Causally debiased time-aware recommendation," in *Proceedings of the ACM on Web Conference 2024*, ser. WWW '24, Singapore, Singapore: Association for Computing Machinery, 2024, 3331–3342, ISBN: 9798400701719. DOI: 10.1145/3589334.3645400. [Online]. Available: https://doi.org/10.1145/3589334.3645400.

[11] Y. Li, W. Chen, and H. Yan, "Learning graph-based embedding for time-aware product recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17, New York, NY, USA: Association for Computing Machinery, 2017, 2163–2166. DOI: 10.1145/3132847.3133060. [Online]. Available: https://doi.org/10.1145/3132847.3133060.

[12] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, vol. 27, no. 1, Aug. 2001. DOI: 10.1146/annurev.soc.27.1.415. [Online]. Available: https://doi.org/10.1146/annurev.soc.27.1.415.

[13] P. V. Marsden and N. E. Friedkin, "Network studies of social influence," *Sociological Methods & Research*, vol. 22, no. 1, Aug. 1993.

[14] A. Singh, S. Fan, and M. Kankanhalli, "Human attributes prediction under privacy-preserving conditions," in *Proceedings of the 29th ACM International Conference on Multimedia*, ser. MM '21, New York, NY, USA: Association for Computing Machinery, 2021, 4698–4706. DOI: 10.1145/3474085.3475687. [Online]. Available: https://doi.org/10.1145/3474085.3475687.

[15] P. Wang, Y. Fu, H. Xiong, and X. Li, "Adversarial substructured representation learning for mobile user profiling," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, New York, NY, USA: Association for Computing Machinery, 2019, 130–138. DOI: 10.1145/3292500.3330869. [Online]. Available: https://doi.org/10.1145/3292500.3330869.

[16] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys '16, New York, NY, USA: Association for Computing Machinery, 2016, 191–198. DOI: 10.1145/2959100.2959190. [Online]. Available: https://doi.org/10.1145/2959100.2959190.

[17] F. Zarrinkalam, H. Fani, and E. Bagheri, "Social user interest mining: Methods and applications," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, 2019, 3235–3236. DOI: 10.1145/3292500.3332279. [Online]. Available: https://doi.org/10.1145/3292500.3332279.

[18] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, ser. WSDM '11, New York, NY, USA: Association for Computing Machinery, 2011, 287–296. DOI: 10.1145/1935826.1935877. [Online]. Available: https://doi.org/10.1145/1935826.1935877.

[19] G. Guo, J. Zhang, and N. Yorke-Smith, "Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proceedings of the AAAI conference on artificial intelligence*, ser. AAAI '15, 2015. DOI: 10.1609/aaai.v29i1.9153. [Online]. Available: https://doi.org/10.1609/aaai.v29i1.9153.

[20] X. Wang, W. Lu, M. Ester, C. Wang, and C. Chen, "Social recommendation with strong and weak ties," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '16, New York, NY, USA: Association for Computing Machinery, 2016, 5–14. DOI: 10.1145/2983323.2983701. [Online]. Available: https://doi.org/10.1145/2983323.2983701.

[21] C. Gao, Y. Zheng, N. Li, *et al.*, "A survey of graph neural networks for recommender systems: Challenges, methods, and directions," *ACM Transactions on Recommender Systems*, vol. 1, no. 1, Mar. 2023. DOI: 10.1145/3568022. [Online]. Available: https://doi.org/10.1145/3568022.

[22] W. Fan, Y. Ma, Q. Li, *et al.*, "Graph neural networks for social recommendation," in *The World Wide Web Conference*, ser. WWW '19, New York, NY, USA: Association for Computing Machinery, 2019, 417–426. DOI: 10.1145/3308558.3313488. [Online]. Available: https://doi.org/10.1145/3308558.3313488.

[23] Y. Tao, Y. Li, S. Zhang, Z. Hou, and Z. Wu, "Revisiting graph based social recommendation: A distillation enhanced social graph network," in *Proceedings of the ACM Web Conference 2022*, ser. WWW '22, New York, NY, USA: Association for Computing Machinery, 2022, 2830–2838. DOI: 10.1145/3485447.3512003. [Online]. Available: https://doi.org/10.1145/3485447.3512003.

[24] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR'19, New York, NY, USA: Association for Computing Machinery, 2019, 235–244. DOI: 10.1145/3331184.3331214. [Online]. Available: https://doi.org/10.1145/3331184.3331214.

[25] S. Wu, J. Chen, T. Xu, *et al.*, "Linking the characters: Video-oriented social graph generation via hierarchical-cumulative gcn," in *Proceedings of the 29th ACM International Conference on Multimedia*, ser. MM '21, New York, NY, USA: Association for Computing Machinery, 2021, 4716–4724. DOI: 10.1145/3474085.3475684. [Online]. Available: https://doi.org/10.1145/3474085.3475684.

[26] W. Guan, X. Song, H. Zhang, M. Liu, C.-H. Yeh, and X. Chang, "Bi-directional heterogeneous graph hashing towards efficient outfit recommendation," in

Proceedings of the 30th ACM International Conference on Multimedia, ser. MM '22, New York, NY, USA: Association for Computing Machinery, 2022, 268–276. DOI: 10.1145/3503161.3548020. [Online]. Available: https://doi.org/10.1145/3503161.3548020.

[27] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "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*, ser. SIGIR'20, New York, NY, USA: Association for Computing Machinery, 2020, 639–648. DOI: 10.1145/3397271.3401063. [Online]. Available: https://doi.org/10.1145/3397271.3401063.

[28] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/2cad8fa47bbef282badbb8de5374b894-Paper.pdf.

[29] Y. Lu, S. Zhang, Y. Huang, *et al.*, "Future-aware diverse trends framework for recommendation," in *Proceedings of the Web Conference 2021*, ser. WWW '21, New York, NY, USA: Association for Computing Machinery, 2021, 2992–3001. DOI: 10.1145/3442381.3449791. [Online]. Available: https://doi.org/10.1145/3442381.3449791.

[30] Q. Tan, J. Zhang, J. Yao, *et al.*, "Sparse-interest network for sequential recommendation," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, ser. WSDM '21, New York, NY, USA: Association for Computing Machinery, 2021, 598–606. DOI: 10.1145/3437963.3441811. [Online]. Available: https://doi.org/10.1145/3437963.3441811.

[31] Z. Xiao, L. Yang, W. Jiang, Y. Wei, Y. Hu, and H. Wang, "Deep multi-interest network for click-through rate prediction," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20, New York, NY, USA: Association for Computing Machinery, 2020, 2265–2268. DOI: 10.1145/3340531.3412092. [Online]. Available: https://doi.org/10.1145/3340531.3412092.

[32] F. Xiao, L. Deng, J. Chen, *et al.*, "From abstract to details: A generative multimodal fusion framework for recommendation," in *Proceedings of the 30th ACM International Conference on Multimedia*, ser. MM '22, New York, NY, USA: Association for Computing Machinery, 2022, 258–267. DOI: 10.1145/3503161.3548366. [Online]. Available: https://doi.org/10.1145/3503161.3548366.

[33] H. Mei and J. M. Eisner, "The neural hawkes process: A neurally self-modulating multivariate point process," in *Advances in Neural Information Processing Systems*, ser. NIPS '17, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/6463c88460bd63bbe256e495c63aa40b-Paper.pdf.

[34] P. Zhao, A. Luo, Y. Liu, *et al.*, "Where to go next: A spatio-temporal gated network for next poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, May 2022. DOI: 10.1109/TKDE.2020.3007194. [Online]. Available: https://doi.org/10.1109/TKDE.2020.3007194.

[35] Q. Shen, S. Zhu, Y. Pang, Y. Zhang, and Z. Wei, "Temporal aware multi-interest graph neural network for session-based recommendation," in *Proceedings of The 14th Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 189, PMLR, 2023. [Online]. Available: https://proceedings.mlr.press/v189/shen23a.html.

[36] L. Huang, Y. Ma, Y. Liu, B. Danny Du, S. Wang, and D. Li, "Position-enhanced and time-aware graph convolutional network for sequential recommendations," *ACM Trans. Inf. Syst.*, vol. 41, no. 1, 2023, ISSN: 1046-8188. DOI: 10.1145/3511700. [Online]. Available: https://doi.org/10.1145/3511700.

[37] F. S. d. Costa and P. Dolog, "Collective embedding for neural context-aware recommender systems," in *Proceedings of the 13th ACM Conference on Recommender Systems*, ser. RecSys '19, Copenhagen, Denmark: Association for Computing Machinery, 2019, 201–209, ISBN: 9781450362436. DOI: 10.1145/3298689.3347028. [Online]. Available: https://doi.org/10.1145/3298689.3347028.

[38] W. Tao, Y. Li, L. Li, *et al.*, "Sminet: State-aware multi-aspect interests representation network for cold-start users recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, ser. AAAI '22, vol. 36, 2022, pp. 8476–8484. DOI: 10.1609/aaai.v36i8.20824. [Online]. Available: https://doi.org/10.1609/aaai.v36i8.20824.

[39] J. McAuley and A. Yang, "Addressing complex and subjective product-related queries with customer reviews," in *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 625–635.

[40] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, Jan. 2017. [Online]. Available: https://arxiv.org/abs/1412.6980.

List of Figures