When Multi-Level Meets Multi-Interest: A Multi-Grained Neural Model for Sequential Recommendation

Yu Tian[1], Jianxin Chang[2], Yanan Niu[2], Yang Song[2], Chenliang Li[1†]
[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, 430072, China
s.braylon1002@gmail.com; cllee@whu.edu.cn
[2]Kuaishou Technology Co., Ltd., Beijing, 10010, China
{changjianxin,niuyanan,yangsong}@kuaishou.com

ABSTRACT

Sequential recommendation aims at identifying the next item that is preferred by a user based on their behavioral history. Compared to conventional sequential models that leverage attention mechanisms and RNNs, recent efforts mainly follow two directions for improvement: *multi-interest learning* and *graph convolutional aggregation*. Specifically, multi-interest methods such as ComiRec and MIMN, focus on extracting different interests for a user by performing historical item clustering, while graph convolution methods including TGSRec and SURGE elect to refine user preferences based on multi-level correlations between historical items. Unfortunately, neither of them realizes that these two types of solutions can mutually complement each other, by aggregating multi-level user preference to achieve more precise multi-interest extraction for a better recommendation. To this end, in this paper, we propose a unified **m**ulti-**g**rained **n**eural **m**odel (named MGNM) via a combination of multi-interest learning and graph convolutional aggregation. Concretely, MGNM first learns the graph structure and information aggregation paths of the historical items for a user. It then performs graph convolution to derive item representations in an iterative fashion, in which the complex preferences at different levels can be well captured. Afterwards, a novel sequential capsule network is proposed to inject the sequential patterns into the multi-interest extraction process, leading to a more precise interest learning in a multi-grained manner. Experiments on three real-world datasets from different scenarios demonstrate the superiority of MGNM against several state-of-the-art baselines. The performance gain over the best baseline is up to 3.12% and 4.35% in terms of NDCG@5 and HIT@5 respectively, which is one of the largest gains in recent development of sequential recommendation. Further analysis also demonstrates that MGNM is robust and effective at user preference understanding at multi-grained levels.

†Chenliang Li is the corresponding author. Work done when Yu Tian was an intern at Kuaishou.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3532081

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Sequential Recommendation, Multi-Interest Learning, Graph Neural Network

ACM Reference Format:
Yu Tian, Jianxin Chang, Yanan Niu, Yang Song, Chenliang Li. 2022. 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 (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3532081

1 INTRODUCTION

With the rapid development of the Internet, recommender systems have become an important tool to solve information overload and enhance competitiveness for many online services such as news feeds, E-commerce, advertising, and social media. Obviously, sequential recommendation, which aims to identify the next item that a user will prefer in terms of her historical behaviors, has drawn increasing attention. The core challenge is how to capture the accurate interests from the user's complex behaviors.

In the past few years, many sequential recommendation solutions have been proposed to model sequential patterns for preference learning. Specifically, earlier works aim to learn a user embedding vector by encoding the user's overall preference from her complex behavior sequence [11, 20, 22, 23, 27]. Typically, a sequence modeling technique is applied over the user behavior sequence. For example, GRU4Rec [11] uses the GRU module to encode preference signals from user behavior sequences. CASER [22] considers the sequence of item embeddings as an image and learns sequential patterns via horizontal and vertical convolutional filters.

Despite the great success achieved by these solutions, all of them ignore the discrimination of different interests by compositing multifaceted preferences into a single vector. Figure 1 illustrates the click sequences of two users from the E-commerce and Micro-video datasets, respectively. Here, each video is displayed by its first frame. From Figure 1(a), in this short click history, there are two main interests: sports and games. To address the above problem, a handful of multi-interest solutions are proposed recently. These methods are devised to learn accurate preference vectors for each user by multi-interest modeling. Generally, a multi-interest network is utilized to explicitly encode the multiple interests according to relevant

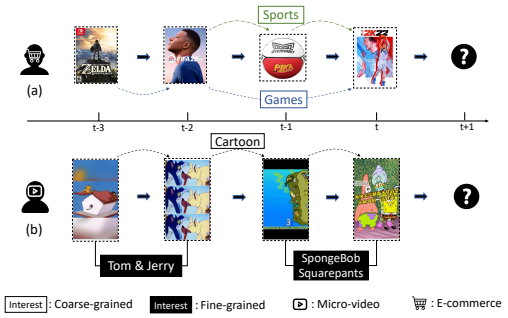


Figure 1: Partial viewing history of two real users in e-commerce and micro-video scenes, respectively. For the user (a), there is a problem that items have an impact on two interests at the same time, i.e. interest overlapping at the $(t-2)$-th and t-th timestamps. For the user (b), there are two different levels of interest in her interaction history: coarse-grained (i.e. Cartoon) and fine-grained (i.e. Tom and Jerry).

information of items in the behavior sequence. For example, MIMN [16] utilizes memory induction units as multiple channels to derive multiple interests from the user's behavior sequence, which delivers large performance gain in the display advertising system of Alibaba. What's more, MIND [13] and ComiRec [1] have been improved respectively on the basis of Capsule Network (CapsNet) [19], and the online system has also gained benefits.

All these multi-interest models, however, take the item as the minimum interest modeling unit, lacking the ability of modeling complex, dynamic and high-order user behaviors. More specifically, as shown in Figure 1 (a), the user mainly focuses on sports (shown in green) and games (shown in blue). Note that the two items in the $(t-2)$-th and t-th timestamps have an impact on the modeling of both two interests (*i.e.,* interest overlapping). In this case, it is difficult to decompose accurately for the existing multi-interest solutions. Moreover, Figure 1 (b) shows that a user's interest would be in different granularities. To address this problem, some efforts propose to combine the sequential modeling with graph neural networks [2, 6]. They build an item graph for the historical interacted items and perform the graph convolution to aggregate the user preference in different levels. However, in comparison to multi-interest solutions, these methods ignore the benefit of multi-interest decomposition. All in all, how to model multiple interests in a multi-grained manner is the problem we want to solve.

To this end, in this paper, we proposed a novel **M**ulti-**G**rained **N**eural **M**odel (named MGNM) via a marriage between multi-interest learning and graph convolutional aggregation. Specifically, MGNM is developed with two major components: *user-aware graph convolution* and *sequential capsule network*. We introduce a learnable process to organize a user's historical items in a user-aware manner, such that the discriminative graph structure and information propagation paths are well uncovered. We then perform graph convolution to derive the item representations iteratively, in which the complex preferences in different levels can be well captured. These multi-level item representations can better reflect the user's diverse preferences. Afterwards, a novel sequential capsule network is proposed to inject the sequential patterns into the multi-interest extraction process, leading to a more precise interest learning. The recommendation is then generated in terms of the relevance between these multiple interests of different levels and the embedding of the candidate item. To summarize, the contributions of this paper are as follows,

- We propose a novel neural model by exploiting the both benefits of multi-interest learning and graph convolutional aggregation for better recommendation performance. Specifically, MGNM can achieve multi-grained user preference learning by integrating multi-level preference composition and multi-interest decomposition into a unified framework.
- We devise a learnable graph construction mechanism to achieve discriminative structure learning over complex user behaviors. Moreover, a sequential capsule network is proposed to exploit temporal information for better multi-interest extraction.
- We conduct extensive experiments on three large-scale datasets collected from real-world applications. The experimental results show significant performance improvements compared with the state-of-the-art technique alternatives. Further analysis is provided to demonstrate the robustness and interpretability of MGNM.

2 RELATED WORK

Considering both sequential modeling and multi-interest learning in recommender systems are two major areas related to our work, we therefore briefly summarize the relevant existing methods in these two areas.

2.1 Sequential Recommendation

Compared with the general recommendation, the scenario of sequential recommendation is different, and its main task is simplified to predict what the user prefers for a commodity pool in the future by using considering the sequential nature of the user historical behaviors. During the early phase, traditional reasoning methods are utilized, such as Markov Chain, which assumes that the next action depends on the previous action sequence. For example, Rendle et al. [17] propose to combine matrix factorization with Markov Chains (MC) to achieve better performance in sequence recommendation. And some works assume that the next action only relies on the last behavior, using first-order Markov chain [4]. Note that these methods are not capable to capture the long-term interests of users effectively due to the limitation of the capability to simulate the dynamic changes of user preferences over time. Then, the emergence of neural networks further enhances recommender systems' ability to extracte the preference of users, so another paradigm of sequence recommendation method based on neural networks in addition to MC-based methods has gradually become the mainstream. The most basic multi-layer perceptions (MLPs) structure extracts the non-linear correlations from user-item interactions [10]. Then a series of models [5, 16, 21, 28] represented by DeepFM [7] are put forward. For the DeepFM model, the FM module is used for a low-order combination of features, and the deep network module is used for the high-order combination of features. By combining the two methods in parallel, the final architecture can learn low-order and high-order combination features at the same time. Referring

to the feature extraction mechanism in texts, audios, and pictures, CNN is used to improve the model capability in sequence recommendation. The CNN architectures are also verified to be effective in this regard to a certain extent, by mapping item sequences to embedding matrices. A representative work is Caser [22], which treats the use's behavior sequence as an "image" and adopts a convolutional neural network to extract user representation. Nevertheless, this mechanism ignores the sequential relations in sequence.

Compared with approaches based on DNN and CNN, RNN is able to capture dynamic time series information [24, 29]. Hidasi et al. [11] first introduce RNN to the sequential recommendation and achieve impressive performance gain over previous methods. Due to the appearance and excellent performance of the RNN network, more and more methods based on the RNN structure are proposed. GRU4Rec [11] first applies Gated Recurrent Units to model the whole session for a more accurate recommendation. To quantify the different importance of past interactions on the next prediction, attention mechanism [23] is adopted. Specifically, attention mechanism makes it easy to memorize various remote dependencies or focus on important parts of the input. In addition, the attention-based methods are often more interpretable [20] than traditional deep learning models. There are some other works that introduce specific neural modules for particular recommendation scenarios, which are mainly based on the combination of RNNs, CNNs, and attention structure, leading to the applications of some emerging network models coming into vogue. For example, memory networks [3, 12], graph neural networks (GNN) [25, 26] that cooperate with the attention mechanism are used to extract short-term features with more consistency or adjacency consideration. SRGNN [25] regards the session history as a directed graph. In addition to considering the relationship between an item and its adjacent previous items, it also considers the relationship with other interactive items. What's more, Fan and Liu et al. [6] integrate the sequence information and collaboration information, use a transformer to capture the temporal relationship in the sequence, and construct a continuous-time bipartite graph. SLi_Rec [27] utilized the fine-grained temporal characteristics of interactive data in the sequence recommendation to stress the ability to modeling sequential behaviors. The recent work represented by TGSRec [6] combines graph and temporal information to further greatly improve the performance of the model.

In a word, most of the existing general sequential approaches are learning to get a single representation of users from an RNN and attention-based model according to the historical behaviors. And graph models, which are capable to aggregate neighbor information, have also been proved to be very effective. Nevertheless, the user history interaction sequence contains more than one discrete interest of the user, and a single vector can not fully express the user preferences. In addition, the noise in the process of graph construction and information aggregation is also an important reason to limit the performance of graph-based sequential models.

2.2 Multi-Interest Recommendation

For a stronger ability to learn the complex behaviors precisely, recently researchers consider that representing user preferences as a single vector is insufficient, more and more sequential recommendation models based on multi-interest, therefore, appear in our field of vision. Li et al. [14] consider that users' interests are dynamic and evolve over time. A pre-trained model based on transformer structure is designed, using the item of the next time step as the label of the interest at the current time step, and then obtains the interest of each time step. The final interest representation is generated by the attentional fusion structure. Pi et al. [15] propose MIMN system which contains modules Neural Turing Machine (NTM), Memory Induction Unit (MIU), etc. In the MIU module, an additional storage unit s is also included, which contains M memory slots. It is considered that each memory slot is a user interest channel. Besides, both MIND [13] and ComiRec [1] devise multi-interest recommendation models on the basis of CapsNet, which uses the idea of neural routing to realize interest decomposition. Note that ComiRec introduces two multi-interest extraction mechanisms including CapsNet and self-attention. At the same time, they also have good applications in the industry. The above methods are multi-interest methods based on sequence models. With the popularity of graph neural networks, the undeniable role of neighbor information has also been proved to be effective obviously. Therefore, the approach of combining graph and multi-interest has also attracted extensive attention in recent years. For example, in SURGE [2], it forms dense clusters in the interest graph to distinguish users' core interests and performs cluster-aware and query-graph graph convolutional propagation to fuse users' current core interests from behavior sequences. These mentioned approaches have also been successfully applied in many recommendation applications and are rather useful and efficient in real-world application tasks.

3 METHOD

In this section, we present the proposed multi-grained neural model in detail. As illustrated in Figure 2, the proposed MGNM consists of two main components: *user-aware graph convolution* and *sequential capsule network*. In the following, we firstly present the formal problem setting. Then, we describe these components, followed by the prediction and model optimization process.

3.1 Problem Formulation

Let $\mathcal{V} = \{x_1, x_2, ..., x_M\}$ denotes the set of all items, $\mathcal{U} = \{u_1, u_2, ..., u_N\}$ be the set of all users, and $\{b_u\}_M^N$ be the behavior sequence set between these N users and M items. Here, for each user u, $b_u = [x_1, x_2, ..., x_m]$ is the sequence of her clicked items following the chronological order, and m is the predefined maximum capacity. The sequential recommendation is to precisely identify the next item x_{m+1} that user u will click in terms of $\{b_u\}_M^N$.

3.2 User-Aware Graph Convolution

In order to extract complex and high-order interests from user click sequences, we consider the graph structure and the aggregation of neighbor information of the target node at different distances in the graph. So at the first step, we convert discrete history behavior into a fully connected item-item graph. Compared with existing methods, we do not artificial use co-occurrence, click of the same user, and other relationships to enhance the graph, because this approach often introduces noise, which affects the performance of

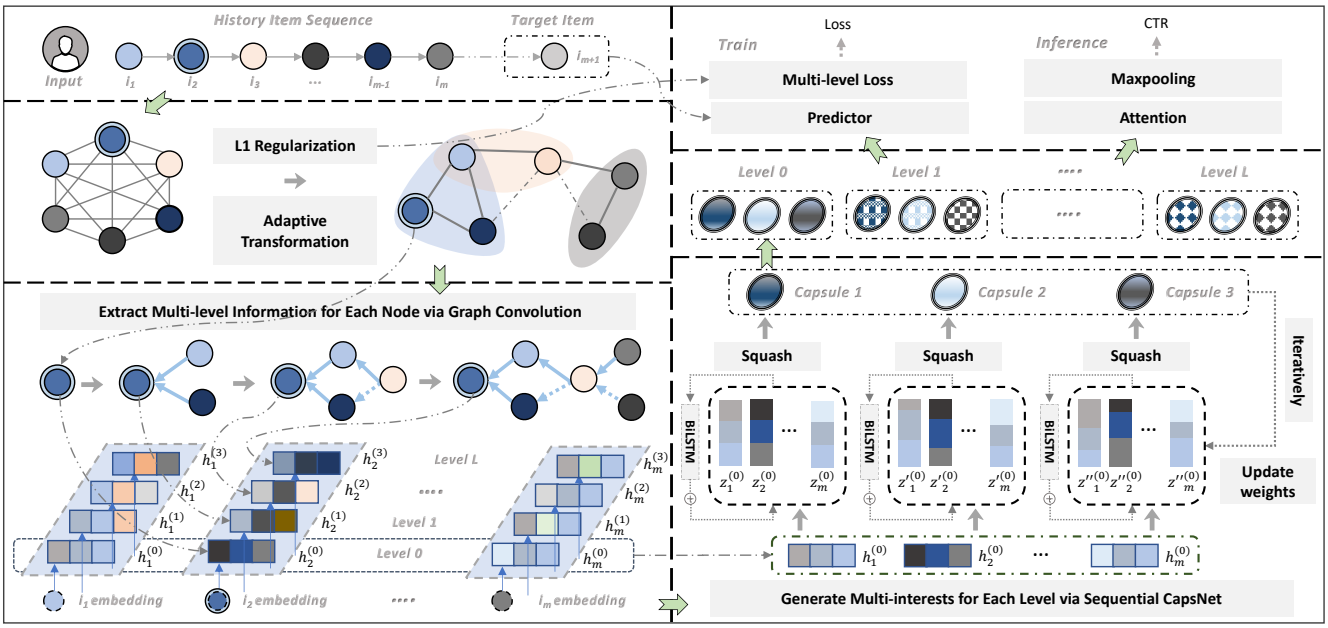


Figure 2: The network architecture of our proposed MGNM. The raw sequence is the historical behavior of a user. By transforming the original sequence into a user-aware adaptive graph and using the neural aggregation function of sequential CapsNet, the timing information is added to the graph in the training process. In the inference stage of the model, the max-pooling layer is used to obtain the final prediction score.

information aggregation in the later convolution process to some extent. In the MGNM, the nodes and users embedding would be updated by using the neural aggregation of CapsNet through gradient feedback and then generate an adaptive graph structure.

3.2.1 Embedding Layer. In the embedding layer, we firstly form a user embedding table $U \in R^{N \times d}$ and an item embedding table $V \in R^{M \times d}$, where d denotes the dimension of the embedding vector. For the given user u and the associated behavior sequence b_u, we can perform the table lookup from U and V to obtain the corresponding user and item embedding representations $\mathbf{x}_u$ and $[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]$ respectively. Hence, the user embeddings U are expected to encode the users' overall preference, while the item embeddings V reflect items' characteristics in this space instead.

3.2.2 Graph Construction. Given the historical behavior sequence $b_u = [x_1, x_2, ..., x_m]$ of user u, we first transform the constituent items into a fully connected undirected graph $\mathcal{G}_u$ by taking each item x_i as a node. It is worth mentioning that we do not condense repeated items in the sequence (*i.e.,* representing multiple clicks of the user), because the multiple clicks of the same item could convey more user preferences. We then introduce $\mathbf{A} \in R^{m \times m}$ to denote the corresponding adjacency matrix, where each entry $\mathbf{A}_{i,j}$ indicates the relatedness between item x_i and item x_j in the perspective of user u. Instead of utilizing behavior patterns to derive matrix $\mathbf{A}$, we choose to learn this relatedness based on their hidden features as follows:

$$\mathbf{A}_{i,j} = sigmoid((\mathbf{x}_i \odot \mathbf{x}_j) \cdot \mathbf{x}_u), \tag{1}$$

where $\odot$ and $\cdot$ denote the Hadamard product and inner product respectively, and *sigmoid* denotes the activation function. We can see that the user embedding $\mathbf{x}_u$ is exploited to achieve user-aware graph construction. That is, the same item pair could have different relatedness values for different users. Also, the use of Hadamard product ensures the symmetry of the adjacency matrix.

Note that graph $\mathcal{G}_u$ is a fully connected. Hence, we need $\mathbf{A}$ to be adequately discriminative to facilitate precise multi-level preference learning. To achieve this purpose, we add $L1$ regularization on the adjacency matrix $\mathbf{A}$ to approximate a certain sparsity.

3.2.3 Graph Convolution. Following the common practice, we perform graph convolution operation over $\mathcal{G}_u$ as follows:

$$\mathbf{H}^{(l+1)} = \delta(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}), \tag{2}$$

$$\tilde{\mathbf{D}}^{-\frac{1}{2}} = \mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}, \tag{3}$$

$$\mathbf{H}^{(0)} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m], \tag{4}$$

where $\mathbf{H}^{(l)}$ denotes the item representations aggregated by the l-th layer ($l \in \{1, \cdots, L\}$), $\delta(\cdot)$ denotes the LeakyReLU nonlinearity, I denotes the identity matrix aiming to add self-loop propagation, $\mathbf{W}$ denotes the trainable parameter and D denotes degree matrix of $\mathbf{A}$. The parameter matrix $\mathbf{W}$ is shared for all L layers. This modification is to facilitate the feature aggregation from the high-order neighbors, which also reduces the model complexity. The item representations composited in each layer can reflect the user's diverse preferences more precisely.

3.3 Sequential Capsule Network

After extracting multi-level item representations $\{\mathbf{H}^{(0)}, \cdots, \mathbf{H}^{(L)}\}$, where $\mathbf{H}^{l} = [\mathbf{h}_1^{(l)}, \cdots, \mathbf{h}_m^{(l)}]$ and $\mathbf{h}_i^{(l)} \in R^d$, we choose to utilize CapsNet to generate the user's multiple interests for each level. Actually, the existing works for multi-interest-based recommendation utilize CapsNet to composite each interest representation through the built-in dynamic routing mechanism. The output of each capsule is equivalent to specific user interests. However, the standard dynamic routing mechanism mainly achieves the function of iterative soft-clustering. It is well validated that temporal information is critical for the sequential recommendation. This is why the application of CapsNet in fine-tuning CTR tasks is limited[1, 13].

Here, we repatch this defect by introducing a sequential encoding layer for CapsNet. Specifically, given the item representations at level l, the ith capsule firstly performs a linear projection over $\mathbf{H}^{(l)}$ as follows:

$$\mathbf{Z}_i = \mathbf{H}^{(l)}\mathbf{W}_i, \tag{5}$$

where $\mathbf{Z}_i = [\mathbf{z}_1^{(l)}, \cdots, \mathbf{z}_m^{(l)}]$ and $\mathbf{W}_i \in R^{d\times d}$ is the trainable parameter for the projection.

We then initialize $\mathbf{g} = [g_1, \cdots, g_m]$ by truncated normal distribution, where g_i is the agreement score indicating the relevance of item x_i towards the capsule. The coupling coefficient $\mathbf{c} \in R^d$ for the corresponding dynamic routing mechanism is then derived via a softmax function:

$$\mathbf{c} = softmax(\mathbf{g}). \tag{6}$$

Then, the capsule derive its output $\mathbf{o}_i$ via a nonlinear squashing function as follows:

$$\mathbf{o}_i = \frac{\|\mathbf{v}_i\|^2}{\|\mathbf{v}_i\|^2 + 1}\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}, \tag{7}$$

$$\mathbf{v}_i = \sum_{j=1}^{m} c_j \mathbf{z}_j^{(l)}, \tag{8}$$

where c_j is the jth element of $\mathbf{c}$. We then update the agreement score g_i as follows:

$$g_i = g_i + \mathbf{o}_i^{\top}\mathbf{z}_i. \tag{9}$$

After this first iteration, we utilize a BiLSTM[1] module to encode sequential patterns and update $\mathbf{Z}_i$ via a residual structure:

$$\mathbf{Z}_i = \mathbf{Z}_i + BiLSTM(\mathbf{Z}_i). \tag{10}$$

We then repeat the above routing process following Equation 6-10 for $\tau - 1$ without further applying BiLSTM encoding over $\mathbf{Z}_i$. That is, total τ iterations are performed for each capsule. The output $\mathbf{o}_i$ in the last iteration is fed into a full-connected layer to derive the ith interest representation $\mathbf{q}_i^{(l)}$ in the l level as follows:

$$\mathbf{q}_i^{(l)} = ReLU(\mathbf{o}_i\mathbf{W}_i'), \tag{11}$$

where $\mathbf{W}_i' \in R^{d\times d}$ is the trainable parameter. Assuming the number of interests is K, we obtain K interest representations $[\mathbf{q}_1^{(l)}, \cdots, \mathbf{q}_K^{(l)}]$ for the lth level. That is, we extract in total $(L+1)\cdot K$ interest representations.

[1]Any other sequential modeling techniques like GRU and Transformer can be straightforwardly applied here.

3.4 Prediction and Model Optimization

3.4.1 Prediction. Given a candidate item x_t, we firstly utilize an attention mechanism to derive the user preference vector $\mathbf{p}_u^{(l)}$ for lth level as follows:

$$\mathbf{p}_u^{(l)} = \sum_{j=1}^{K} a_j \mathbf{q}_j^{(l)}, \tag{12}$$

$$a_j = \frac{\exp(\mathbf{q}_j^{(l)\top}\mathbf{x}_t)}{\sum_{k=1}^{K}\exp(\mathbf{q}_k^{(l)\top}\mathbf{x}_t)}, \tag{13}$$

where a_j is the attention weight for jth interest. Then, we choose inner product to calculate the recommendation score as follows:

$$\hat{y}_{u,i}^{(l)} = \mathbf{p}_u^{(l)\top}\mathbf{x}_t, \tag{14}$$

where $\hat{y}_{u,i}^{(l)}$ denotes the recommendation score for the lth level. Note that different users could have different interest granularity. In other words, some users' interests are very complex and dynamic, the high-level user preference is more accurate. On the other hand, some users' interests are simple and straightforward, it is more appropriate to utilize the low-level user preference or even original item representations. Hence, we derive the final recommendation score by using the max-pooling:

$$\hat{y}_{u,i} = \max(\hat{y}_{u,i}^{(0)}, \cdots, \hat{y}_{u,i}^{(L)}). \tag{15}$$

3.4.2 Model Optimization. For the sake of enabling the model to capture user interests of different granularity from low-level to high-level, we choose to define a cross-entropy loss for each level. Thus, the final loss is formulated as follows:

$$\mathcal{L}_{all} = \sum_{l=0}^{L}\mathcal{L}_l + \theta_1\mathcal{L}_1 + \theta_2\mathcal{L}_2, \tag{16}$$

$$\mathcal{L}_l = -\sum_{u,i}[y_{u,i}ln(\hat{y}_{u,i}^{(l)}) + (1-y_{u,i})ln(1-\hat{u}_{u,i}^{(l)})], \tag{17}$$

where $y_{u,i}$ denotes the ground truth for user u and item x_i, $\mathcal{L}_1$ denotes the L_1 norm of the matrix $\mathbf{A}$, $\mathcal{L}_2$ denotes the L_2 norm of all model parameters, θ_1 and θ_2 denote the hyperparameters.

4 EXPERIMENTS

In this section, we conduct extensive experiments on three real-world datasets from different domains for performance evaluation. We then analyze the contributions of several components and different settings for MGNM[2]. Finally, a thorough analysis of ablation experiments and a framework optimizer exploration are presented.

4.1 Experimental Settings

Datasets. The first dataset (namely *Micro-video*) is collected from a leading large-scale Micro-video sharing platform. This dataset contains 60, 813 users and their interaction records over seven days (*i.e.,* October 22 to October 28, 2020). We take the interactions made in the first six days as the training set. The interactions were made before 12PM on the last day as the validation set, and the rest as the test set.

[2]The code implementation is available at https://github.com/WHUIR/MGNM

Table 1: Statistics of the three datasets.

Datasets	#Users	#Items	#Interactions
Micro-video	60,813	292,286	14,952,659
Musical Instruments	60,739	56,301	946,627
Toys and Games	313,557	241,657	6,212,901

The other two datasets are from Amazon product datasets[3]: *Musical Instruments* and *Toys and Games*. Here, each user interaction in the Amazon dataset is associated with a user rating score. Following the previous works [8, 9, 18], we take each user interaction with a rating score larger than 2 as the positive. We then organize these interactions and split the interaction sequence with the ratio of 7:1:2 to form the training, validation, and test set respectively following the chronological order. We further remove users whose length of history sequence is 1.

Table 1 summarizes detailed statistics of the three datasets after preprocessing. The Micro-video dataset includes a large number of items, while Toys and Games is much smaller. Also, the Musical Instruments is the smallest according to the interaction number. We can see that these three real-world datasets hold different characteristics, covering a broad range of real-world scenarios.

Baselines. We compare the proposed MGNM against the following state-of-the-art sequential recommendation methods:

- **Caser** [22] is A CNN-based model which applies horizontal and vertical convolutional filters to capture the point-level, union-level, and skipping patterns for sequential recommendation.
- **A2SVD** [27] is short for the asynchronous SVD method, which modifies the prediction model to express the user as the superposition of items. Combined with implicit feedback data, the parameters of the model are reduced, and the interpretability of the original SVD model is enhanced.
- **GRU2Rec** [11] utilizes the gated recurrent unit to model the session sequence for recommendation.
- **SLi_Rec** [27] improve the traditional RNN structure by proposing a temporal-aware controller and a content-aware controller so that contextual information can guide the state transition. An attention-based framework is proposed to combine the user's long-term and short-term preferences. Hence, the representations can be generated adaptively according to the specific context.
- **MIMN** [15] is a state-of-the-art multi-interest model that utilizes a multi-channel memory metwork, to capture user interests from the sequential behaviors .
- **MIND** [13] is a multi-interest learning model that utilizes CapsNet to capture diverse interests of a user.
- **ComiRec** [1] is a recent multi-interest model containing a multi-interest module and an aggregation module. The multi-interest module captures a variety of interests from the user behavior sequence, and can retrieve the candidate item set in a large-scale item pool. Then the aggregation module uses controllable factors to balance the accuracy and diversity for recommendation. Two variants of ComiRec are used for performance comparison: ComiRec-DR and ComiRec-SA, where CapsNet and self-attention are used for multi-interest extraction respectively.
- **SURGE** [2] is a uptodate graph neural model for sequential recommendation, which performs cluster-aware and query-graph propagation to fuse users' current core interests from behavior sequences.
- **TGSRec** [6] is also a uptodate graph neural model that considers temporal dynamics inside the sequential patterns.

All these baselines can be divided into four categories: **(1)** traditional sequential models that utilize RNN and attention mechanism (*i.e.,* Caser, A2SVD and GRU4Rec); **(2)** temporal-aware models that exploit the timestamp information (*i.e.,* SLi_Rec and TGSRec); **(3)** multi-interest models that derive various user interest (*i.e.,* MIMN, MIND, ComiRec-DR, ComiRec-SA and SURGE); **(4)** graph nerual models that exploit high-order correlations (*i.e.,* SURGE).

Hyperparameter Settings. For a fair comparison, all methods are implemented in Tensorflow and learnt with Adam optimizer. The learning rate, mini-batch size are set to $1e-3$ and 256. The number of negative samples is 5 in the training stage for all three datasets. We tuned the parameters of all methods over the validation and set the embedding size as 16 and 40 for Amazon datasets and Micro-video datasets respectively. Specifically, as to MGNM, we found our model performs relatively stable when $K = 4$, $L = 3$, and $\theta_1 = 1e-6, \theta_2 = 1e-5$.

Evaluation Metric. Following the same setting in [6], we sample $1,000$ negative items for each testing instance. Four common metrics: hit rate (HR), mean reciprocal rank (MRR), and normalized discounted cumulative gain (NDCG) and Group AUC (GAUC), are used for performance evaluation. For method, we repeat the experiment 5 times and report the average results. The statistical significance test is conducted by the student's $t\text{-}test$.

4.2 Performance Evaluation

The overall performance of all methods is reported in Table 2. Here, we make the following observations.

As for traditional sequential models that utilize RNN and attention mechanism, they are difficult to achieve better performance. Compared with temporal-aware models and multi-interest models, they are not suitable for complex and various user interest modeling. The temporal-aware models perform very well in Amazon datasets. Specifically, on the Music Instruments dataset, TGSRec and Sli_Rec achieve the best performance in terms of GAUC and NDCG@5 respectively. They also achieve strong performance in terms of the other six metrics for both Toys and Games and Music Instruments. It is worthwhile to mention that TGSRec needs to build a global graph and takes the interactions at different time points as edges. This design choice requires much more computation cost for graph retrieval and convolution. Note that the graph constructed on the Micro-video dataset contains more than 200 million edges. We utilize the implementation released by the original authors for evaluation. The time of an epoch training exceeds $1,200$ hours. Hence, we do not obtain results on the Micro-video dataset.

Also, given the superiority of these temporal-aware models on both Amazon datasets, the multi-interest models perform better on

[3]http://snap.stanford.edu/data/amazon/

Table 2: Performance comparison of different methods across the three datasets. The best and second-best results are highlighted in boldface and underlined respectively. $*$ indicates that the performance difference against the best result is statistically significant at 0.05 level. Note that TGSRec took too long to train hence has no results on the large Micro-video dataset. See context for details.

Method	Micro-video				Toys and Games				Music Instruments			
	GAUC	NDCG@5	HIT@5	MRR@5	GAUC	NDCG@5	HIT@5	MRR@5	GAUC	NDCG@5	HIT@5	MRR@5
Caser	0.6917*	0.0964*	0.1417*	0.0815*	0.6234*	0.0679*	0.1012*	0.0569*	0.6763*	0.0955*	0.1178*	0.0883*
A2svd	0.6808*	0.0443*	0.0686*	0.0364*	0.6846*	0.0507*	0.0739*	0.0430*	0.6652*	0.0956*	0.1368*	0.0820*
GRU4Rec	0.6944*	0.0702*	0.1050*	0.0589*	0.6624*	0.0840*	0.1278*	0.0697*	0.6498*	0.0619*	0.1049*	0.0478*
SLi_rec	0.6903*	0.0948*	0.1390*	0.0802*	0.7847*	0.0932*	0.1327*	0.0803*	0.6912*	**0.1078**	0.1507*	0.0937*
TGSRec	–	–	–	–	0.7915*	0.1410*	0.2027*	0.1164*	**0.7759**	0.0946*	0.1653	0.0729*
MIMN	0.7387*	0.1151*	0.1683*	0.0977*	0.7224*	0.1158*	0.1676*	0.0988*	0.6787*	0.0955*	0.1509*	0.0750*
MIND	0.6778*	0.08582*	0.1367*	0.0700*	0.6611*	0.1015*	0.1510*	0.0824*	0.6588*	0.1040*	0.1422*	0.0898*
ComiRec-DR	0.7028*	0.0863*	0.1307*	0.0718*	0.6681*	0.1131*	0.1597*	0.0978*	0.6647*	0.1091*	0.1541*	0.0943*
ComiRec-SA	0.6249*	0.0354*	0.0577*	0.0281*	0.6486*	0.0665*	0.0977*	0.0563*	0.6559*	0.0820*	0.1204*	0.0694*
SURGE	0.8116*	0.1091*	0.1728*	0.0883*	0.7863*	0.0930*	0.1353*	0.0791*	0.6902*	0.1056*	0.1494*	0.0913*
MGNM	**0.8325**	**0.1463**	**0.2163**	**0.1232**	**0.8078**	**0.1611**	**0.2231**	**0.1408**	0.7480*	0.1057	**0.1658**	**0.1021**

the Micro-video dataset. This suggests that neither temporal-aware models nor multi-interest models are robust enough to achieve precise preference understanding across different scenarios. Considering the semantic space in the Micro-video recommender scenario could be much broader than commodities in E-Commerce scenarios, the interest of each user also becomes more complex. It is reasonable that the multi-interest models could achieve better recommendation performance instead.

Our proposed MGNM has obvious improvement in most settings for the three datasets including Micro-video, Toys and Games, and Music Instruments. In detail, MGNM performs significantly better than all the baselines on 10 out of 12 dataset and metric combinations. Although our MGNM achieves only comparable NDCG@5 performance against SLi_Rec and performs a bit worse to TGSRec in terms of GAUC both on the Music Instruments dataset, we need to emphasize that both SLi_Rec and TGSRec exploit additional timestamp features to seize more discriminative capacity. In other words, our model MGNM lacks one-dimensional timestamp characteristics than these two models (i.e. SLi_Rec and TGSRec). Note that modeling multi-grained multi-interest in MGNM and exploiting timestamp information are not mutually excluded. As shown in Equation 10, it is straightforward to include fine-grained timestamp information in the sequential capsule network component[4]. Moreover, we can see that our proposed MGNM performs increasingly better on larger datasets. The relative performance gain by MGNM against the best baseline is in the range of $1.63\% - 2.44\%$ and $2.09\% - 4.35\%$ on Toys and Games and Micro-video datasets respectively. This further confirms that our MGNM is effective to capture multi-grained user interests for large-scale real-world scenarios that are rich in semantics.

4.3 Model Analysis

Here, we investigate the impact of each design choice and important parameter settings to the performance of MGNM.

Ablation Study. We conduct an ablation study for each design choice in MGNM to justify their validity. Specifically, these factors include user-aware graph convolution (UGCN), the $L1$ regularization on the adjacency matrix A ($L1$Norm), sequential capsule network without sequential encoding layer (BiLSTM), and the max-pooling based prediction (MaxPool). As to the sequential capsule network, we also examine the following variants:

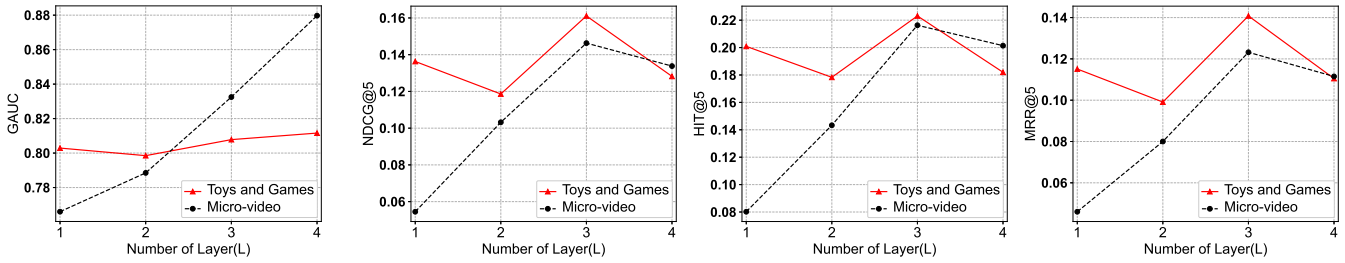
- SCN→ BiLSTM: We replace the sequential capsule network with BiLSTM to encode the user behavior sequence in different levels. The last hidden state generated by the BiLSTM is taken the user interest in the corresponding level.
- SCN→ SumPool: We replace the sequential capsule network with a sum pooling mechanism. Similar to SCN→ BiLSTM, the resultant representation is taken the user interest for the corresponding level.
- SCN→ SelfAtt: We replace the sequential capsule network with a self-attention mechanism. The candidate item is utilized to derive the user interest for each level by using an attention mechanism.
- SCN (Transformer): We replace the built-in BiLSTM in the sequential capsule network with a powerful transformer module.

Table 3 reports the performance of these variants and the full MGNM model on Toys and Games dataset[5]. Here, we can make the following observations.

Firstly, The L_1 regularization indeed improves the discriminative capacity of user-aware graph convolution. The experimental results show that the performance degradation without it is obvious. When we remove the multi-level item representation learning supported by user-aware graph convolution (*i.e.,* w/o UGCN), substantial performance degradation is also experienced by MGNM, which illustrates the effectiveness of user-aware graph convolution and multi-level preference learning significantly. Also, we can find

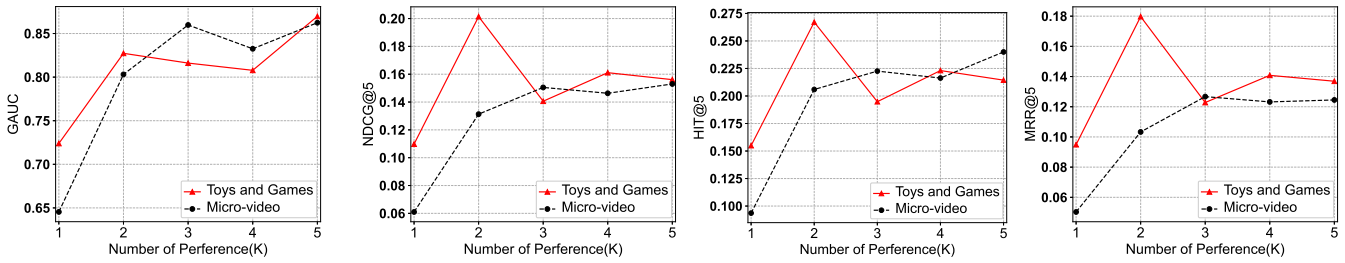
[4]We leave it as a part of our future work.

[5]Similar observations are also made in the other two datasets



(a) GAUC **(b) NDCG@5** **(c) HIT@5** **(d) MRR@5**

Figure 3: The performance of different L values on Toys and Games and Micro-video Datasets.



(a) GAUC **(b) NDCG@5** **(c) HIT@5** **(d) MRR@5**

Figure 4: The performance of different K values on Toys and Games and Micro-video Datasets.

Table 3: The ablation study of MGNM on Toys and Games Dataset. The best results are highlighted in boldface.

Model	Toys and Games			
	GAUC	NDCG@5	HIT@5	MRR@5
w/o UGCN	0.7499	0.0929	0.1325	0.0799
w/o $L1$Norm	0.7757	0.1306	0.1848	0.1128
w/o BiLSTM	0.6743	0.1205	0.1689	0.1046
w/o MaxPool	0.8491	0.0980	0.1430	0.0832
SCN→ BiLSTM	0.6589	0.0838	0.1223	0.0712
SCN→ SumPool	0.6651	0.0846	0.1232	0.0720
SCN→ SelfAtt	0.6724	0.0791	0.1148	0.0674
SCN (Transformer)	0.6663	0.0923	0.1321	0.0792
MGNM	**0.8078**	**0.1611**	**0.2231**	**0.1408**

that MGNM experiences a large performance degradation by removing the sequential encoding layer (*i.e.,* w/o BiLSTM). This is reasonable since the sequential patterns have been well validated to be effective for the sequential recommendation. Now, we further validate that sequential patterns are also very useful for multi-interest learning. At last, the max-pooling-based prediction plays a great role in improving all four performance metrics. As we described earlier, the max-pooling mechanism is flexible in capturing the complex user preference from multi-grained interests.

Secondly, we further dive deep into the effectiveness of the sequential capsule network component. The first three variants (*i.e.,* SCN→ BiLSTM, SCN→ SumPool, SCN→ SelfAtt) aim to remove the multi-interest modeling by considering only the multi-level user preferences. We can see that these three variants all experience significant performance degradation across the four metrics. Recall that MGNM w/o UGCN also produces a substantial performance degradation above. These two observations suggest that both user-aware graph convolution and sequential capsule network works as a whole and either of them complements the other, leading to better user preference understanding. At last, we find that encoding sequential patterns with a heavy module like transformer achieves better performance than SCN→ BiLSTM, SCN→ SumPool, and SCN→ SelfAtt in terms of NDCG@5, HIT@5 and MRR@5. This also validates the benefit of modeling sequential patterns for multi-interest learning. However, the huge number of parameters involved in a transformer module could complicate the model optimization process. Note that we also derive multi-level item representations by the user-aware graph convolution, a lightweight sequential model like BiLSTM is sufficient for the next step. At the same time, to the best of our knowledge, there are no previous methods to integrate sequential modeling with CapsNet.

Impact of L Value. Recall that we stack L layers of graph convolution in MGNM to reflect the user's diverse preferences in multi-grained manner. A larger L value can recruit more high-order neighbors to derive the user's preference more and more distant neighbor information is aggregated. However, some noisy information would also be included to deliver adverse impact. In Figure 3, we plot the performance patterns of varying L values for both Toys and Games and Micro-video datasets. It is reasonable to see that all NDCG@5, HIT5 and MRR@5 scores firstly increase when L becomes large ($L \leq 3$), and then decrease when L is too

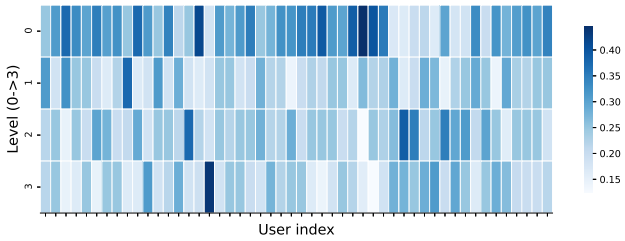


Figure 5: Visualization of multi-level user interest distribution on Micro-video dataset (*Best viewed in color*).

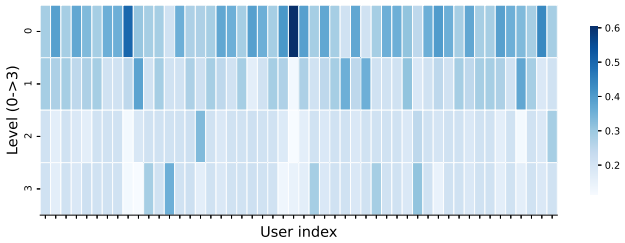


Figure 6: Visualization of multi-level user interest distribution on Toys and Games dataset (*Best viewed in color*).

large ($L > 3$). Also, the metric GAUC seems to be very stable for different L values.

Impact of K Value. The number of interests K in MGNM controls the diversity of user preferences. Figure 4 plots the performance patterns of varying K values for both Toys and Games and Micro-video datasets. We can observe that a single interest representation (*i.e.,* $K = 1$) achieves the worst performance across the four metrics. The optimal K value is 2 and 4 for Toys and Games and Micro-video datasets respectively. Moreover, we can see that MGNM achieves relatively more stable performance when K is in the range of $[3, 5]$. This is reasonable since the semantic space of the Micro-video dataset is much broader than the Toys and Games dataset.

Multi-Level User Interest Distribution. In Figure 3, we examine the impact of different L layers in Micro-video. Here, we further investigate whether the multi-level user preferences indeed perform different roles for different users. Specifically, we randomly sample 50 users from the Micro-video and Toys and Games datasets, respectively. For each user, we include her positive items in the test set and thousands of random negative items, and count the activated preference level by the max-pooling based predictor (ref. Equation 15). Figure 5 and Figure 6 plots the distribution of these activated levels for each user on Micro-video and Toys and Games datasets, respectively. We can observe that the desired preference level is quite different for different users. Also, the first two layers are adequate for most users in MGNM on Toys and Games dataset. But we also need to derive high-level preferences for a few users (*i.e.,* $L \geq 2$) in Figure 6. As for the Micro-video dataset with a larger semantic space, the role of high-level preferences becomes more significant to all users from Figure 5. On the whole, users have more high-level preferences on Micro-video. In other words, users' interests in Micro-video scenes are higher-level, more complex, and change faster, which we mentioned in figure 1 and the previous analysis. Thus, this phenomenon is in line with our expectations, which well proves the effective impact of the multi-level mechanism. Furthermore, in the inference stage, we replace max-pooling with sum-pooling to further verify the influence of max-pooling structure in Figure 7. In combination with the distribution of user interests in Figure 5 and 6) and the better experimental results of max-pooling than sum-pooling in Figure 7, this suggests that the user-aware graph convolution does distinguish the user's interest in multi-level precisely and the multi-level mechanism does promote the performance.

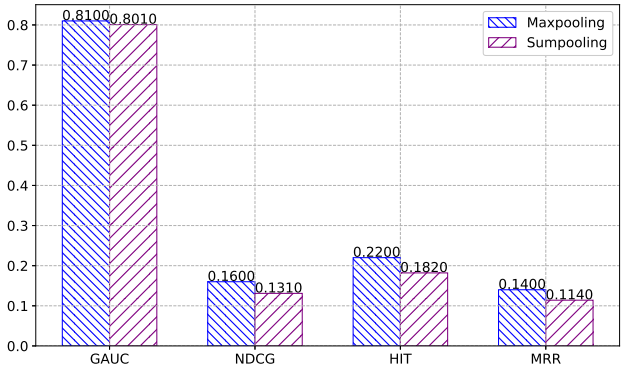


Figure 7: Max-pooling vs. sum-pooling for MGNM in the inference stage.

Table 4: Runtime comparisons for different datasets.

Datasets	Per Iteration (s)	Iterations	Total Time (m)
Micro-video	0.3825	15,311	97.60
Toys and Games	0.1843	13,202	40.55
Music Instruments	0.0598	2,373	2.37

Time Complexity Analysis. Table 4 reports the runtime of MGNM training procedure for a single user on different datasets by using a single GPU. Although the MGNM adopt the graph convolution, we can see that the model training with 15M interactinos takes about 1.5H for one epoch, which is computationally efficient.

5 CONCLUSION

In this paper, we proposed a novel **m**ulti-**g**rained **n**eural **m**odel (named MGNM) with a combination of multi-level and multi-interest as a unified solution for sequential recommendation task. A learnable process was introduced to re-construct loose item sequences into tight item-item interest graphs in a user-aware manner. We then performed graph convolution to derive the item representations iteratively, in which the complex preferences in different levels can be well captured. Afterwards, a novel sequential CapsNet was designed to inject the sequential patterns into the multi-interest extraction process, leading to a more precise interest modeling. Extensive experiments on three real-world datasets in different recommendation scenes demonstrated the effectiveness of the multi-level

and multi-interest mechanisms. Further studies on the number of preferences and multi-level user interest distribution confirmed that our method was able to deliver recommendation interpretation at multi-level granularities.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (No. 61872278); and Young Top-notch Talent Cultivation Program of Hubei Province. Chenliang Li is the corresponding author.

REFERENCES

[1] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.

[2] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential Recommendation with Graph Neural Networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 378–387.

[3] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 108–116.

[4] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Twenty-Third international joint conference on Artificial Intelligence*.

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[6] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. 2021. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 433–442.

[7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[8] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 161–169.

[9] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining*. 191–200.

[10] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.

[11] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[12] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 505–514.

[13] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.

[14] Shihao Li, Dekun Yang, and Bufeng Zhang. 2020. MRIF: Multi-resolution Interest Fusion for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1765–1768.

[15] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.

[16] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 1149–1154.

[17] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. 995–1000.

[18] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[19] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules. In *In the 31th Conference on Neural Information Processing Systems*. 3856–3866.

[20] Ying Sha and May D Wang. 2017. Interpretable predictions of clinical outcomes with an attention-based recurrent neural network. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. 233–240.

[21] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 255–262.

[22] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 565–573.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[24] Yaqing Wang, Caili Guo, Yunfei Chu, Jenq-Neng Hwang, and Chunyan Feng. 2020. A cross-domain hierarchical recurrent model for personalized session-based recommendations. *Neurocomputing* 380 (2020), 271–284.

[25] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.

[26] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.

[27] Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. 2019. Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation. In *In the 30th International Joint Conference on Artificial Intelligence*. 4213–4219.

[28] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *The 38th European Conference on Information Retrieval*. 45–57.

[29] Yuwen Zhou, Changqin Huang, Qintai Hu, Jia Zhu, and Yong Tang. 2018. Personalized learning full-path recommendation model based on LSTM neural networks. *Information Sciences* (2018), 135–152.