Hindawi
International Journal of Intelligent Systems
Volume 2024, Article ID 4169402, 19 pages
https://doi.org/10.1155/2024/4169402

Research Article

A Recommendation Approach Based on Heterogeneous Network and Dynamic Knowledge Graph

Shanshan Wan,[1,2] **Yuquan Wu**,[1] **Ying Liu**,[3] **Linhu Xiao**,[4] **and Maozu Guo**[1,2]

[1] *School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China*
[2] *Beijing Key Laboratory of Intelligent Processing for Building Big Data, Beijing 102616, China*
[3] *People's Bank of China, Lanzhou, Gansu 730000, China*
[4] *Lanzhou Jiaotong University, Lanzhou, Gansu 730070, China*

Correspondence should be addressed to Maozu Guo; guomaozu@bucea.edu.cn

Received 24 March 2023; Revised 25 September 2023; Accepted 12 October 2023; Published 3 January 2024

Academic Editor: Yu-an Tan

Besides data sparsity and cold start, recommender systems often face the problems of selection bias and exposure bias. These problems influence the accuracy of recommendations and easily lead to overrecommendations. This paper proposes a recommendation approach based on heterogeneous network and dynamic knowledge graph (HN-DKG). The main steps include (1) determining the implicit preferences of users according to user's cross-domain and cross-platform behaviors to form multimodal nodes and then building a heterogeneous knowledge graph; (2) Applying an improved multihead attention mechanism of the graph attention network (GAT) to realize the relationship enhancement of multimodal nodes and constructing a dynamic knowledge graph; and (3) Leveraging RippleNet to discover user's layered potential interests and rating candidate items. In which, some mechanisms, such as user seed clusters, propagation blocking, and random seed mechanisms, are designed to obtain more accurate and diverse recommendations. In this paper, the public datasets are used to evaluate the performance of algorithms, and the experimental results show that the proposed method has good performance in the effectiveness and diversity of recommendations. On the MovieLens-1M dataset, the proposed model is 18%, 9%, and 2% higher than KGAT on $F1$, NDCG@10, and AUC and 20%, 2%, and 0.9% higher than RippleNet, respectively. On the Amazon Book dataset, the proposed model is 12%, 3%, and 2.5% higher than NFM on $F1$, NDCG@10, and AUC and 0.8%, 2.3%, and 0.35% higher than RippleNet, respectively.

1. Introduction

In recent years, the continuous development of the Internet and big data has led to the rapid growth of network resources. Information explosion and knowledge scarcity coexist. How to quickly find resources which match users' needs from massive data has become a hot topic. To provide users with the most appropriate resources, recommender systems have been popular in various scenarios, such as news recommendations [1, 2] POI location recommendation [3, 4], recommendation of goods [5], and learning resource recommendation [6].

Cold start [7, 8] and data sparsity [9, 10] are the main problems faced by recommender systems. The applications of context information, social networks, hybrid algorithms, and other methods have greatly addressed traditional cold start and data sparsity problems. Some studies added auxiliary information to the knowledge graph (KG) to achieve accurate recommendations [11, 12]. Some studies utilized RippleNet to extract user characteristics and expand user preferences [13]. In order to represent the potential preference behavior of users more accurately and comprehensively, how to make full use of auxiliary information needs to be studied further.

In addition, popular items are usually overrecommended in recommender systems, which will lead to various data bias problems for items. Data bias includes exposure deviation, selection deviation, popularity deviation, circular deviation, and consistency deviation. The existence of data bias makes popular items over-recommended, and items that may be of interest to users are ignored. This leads to a decrease in the freshness and a lack of diversity of recommendations, ultimately having a negative impact on users and product providers. Therefore, it is very important to properly mitigate and handle the bias problem [14, 15]. For both selection bias and exposure bias, common solutions include uniform data, inverse tendency score, heuristic confidence weight, and sampling. These methods are effective in improving the precision of the recommender system, but the accuracy is insufficient. Moreover, these methods depend heavily on expertise.

In this study, heterogeneous network graphs and dynamic knowledge graphs are designed for multidimensional user feature extraction [16]. Heterogeneous network is a special information network which contains multiple types of nodes [17–19]. The node type of heterogeneous network can be multimodal, so the heterogeneous network can retain more comprehensive semantic and structural information, which helps capture the implicit relationship between items and reduce the dependence of recommender systems on rating data. Heterogeneous network can help complete the user preferences and user relationships, and effectively address the cold start and data sparsity problems. The completion of entities and entity relationships based on heterogeneous networks will help reduce the impact of exposure bias [20, 21] and selection bias [22, 23].

In addition, most of the existing research focuses on user's long-term preferences. User's long-term preferences are usually related to their inherent characteristics and can be extracted from a large number of users' behaviors in a certain period of time. However, with the rapid development of Internet, the trend of popularity has changed greatly. User preferences are often changing due to public opinion, frequent online communication, and other unexpected events. Therefore, the short-term preferences of users also provide potential possibilities to develop long-term preferences. If we pay attention to the short-term preferences of users, the recommendation results can be more accurate and diversified.

In order to extract user features and mine potential user preferences, attention mechanisms are widely used in recommender systems [24–26]. Graph attention network (GAT) is different from some previous graph neural networks based on spectral domain. It can aggregate neighbor nodes through the attention mechanism to achieve adaptive distribution of weights of different neighbor nodes. It has advantages such as high efficiency and portability [27]. For example, the graph attention network can complete the relationship weight between different nodes and improve the accuracy of recommendations according to users' intimacy and the interaction behavior of users participating in different activities [28, 29].

Based on the above studies, we propose an approach to build heterogeneous networks for recommendations. First, a heterogeneous network diagram with multimodal nodes is established based on cross-domain multi-platform information, from which the implicit preference of users can be extracted. Data extraction is performed on heterogeneous multimodal node graphs to construct basic knowledge maps. Second, the timeliness of user preferences is considered, and a time warehouse triggering mechanism is set up. At the same time, the graph attention network component is used to extract the short-term characteristics of users. Finally, according to the attention weight function, the improved RippleNet is used to calculate the click probability [22]. Thus, the accuracy and diversity of recommendations could be further improved.

The rest of this paper is organized as follows: Section 2 introduces the research of heterogeneous networks, graph attention networks (GAT), and RippleNet. Section 3 details the proposed recommender system based on dynamic knowledge graphs in heterogeneous networks. Section 4 carries out experiments and analyzes the experimental results. Section 5 summarizes the research of this study and introduces the future work.

2. Related Work

This section focuses on the study of feature extraction and relationship enhancement for users and items in recommender systems and summarizes the research of heterogeneous networks, graph attention network, and RippleNet. The motivation of our study is also introduced.

2.1. Construction and Application of Heterogeneous Networks. There is much research focusing on cold start and data sparsity of recommender systems, such as using knowledge graphs, deep learning, and hybrid recommendation methods. To address the problem of selection bias [30, 31], the most common scheme is data filling and tendency score. Exposure deviation [14, 20] can be addressed by using heuristic weight, uniform data, and negative sample sampling methods, but these methods depend too much on human's professional experience and they are insufficient in high recommendation accuracy.

Heterogeneous networks can not only integrate different types of objects and their interactions but also integrate information from heterogeneous data sources [32–35]. Heterogeneous network has become an effective information modelling method for it contains multiple types of nodes and multiple types of edges [36]. Wang et al. proposed an extensible dimension recommendation model based on heterogeneous network in view of the current lack of label models that simultaneously consider multidimensional information [37]. Their method can be applied to different dimension label data to recommend labels for users at the same time, but its real-time performance and recommendation efficiency need to be improved. Shi et al. proposed a recommendation method based on heterogeneous information network (HIN) for heterogeneous network embedding [17]. They aimed at the difficult modelling of complex auxiliary information and the problem of data cold start. Their method effectively uses the auxiliary information in heterogeneous networks and designs a random walking strategy based on meta path to obtain a more meaningful network embedded node sequence. However, the method needs further improvement in the effectiveness of auxiliary data selection. Hu et al. proposed a deep neural network model of common attention mechanism based on heterogeneous information networks to solve the impact between meta paths and related user pairs in interaction [38]. However, the weight of edges in heterogeneous information networks is set as the same, and the different weights of different types of edges are not taken into account, which makes the recommendation result not ideal. Many studies also proposed solutions based on heterogeneous information networks in the field of recommendation, but these models failed to effectively consider the differences between different meta paths [39–41].

The above research studies completed the missing information to a certain extent by establishing various heterogeneous graphs and also improved the accuracy of the recommender system. However, these research did not use users' multi-platform behavior to complete the information. In this study, we integrate the interaction behavior of multimodal nodes and establish virtual nodes and virtual relationships to mine users' implicit preferences. Thus, the real-time diversity of the recommender system is expected to be improved.

2.2. Current Status of Graph Attention Application. In 2018, Veličković et al. proposed a graph attention network for graph structure data [27]. The graph attention network uses the self-attention method to calculate the attention of a node in the graph relative to each adjacent node. The graph attention network is widely used in text classification, software detection, and other fields and has achieved good results. Its application in recommender systems has also become a popular topic [42]. The graph attention network does not need to pay attention to the whole graph structure and it can give different weights of neighbor nodes.

Wang et al. proposed the knowledge graph attention network (KGAT) to solve the problem of ignoring the relationship between data in some models [42]. It explicitly models the high-order connectivity in knowledge graph (KG) in an end-to-end manner and uses the attention mechanism to distinguish the importance of neighbors. However, this method does not consider the situation that the relationship will change over time, and it has poor real-time performance. Wang et al. proposed a relational metric social recommendation model based on graph attention network to solve the problem of excessive interference information in recommender systems [43]. Dual graph attention network is designed in the item domain and social domain to adaptively aggregate domain characteristics of users or items. The complex interactions between corresponding neighbors are modelled as relation vectors by using multilayer neural networks. However, this method lacks in modelling social relationships of different types or strengths. Zeng and Liu proposed a model which combines knowledge graph and graph attention network and adds an interest evolution module to graph attention network to capture user interest changes and generate Top-N recommendations [44]. However, the robustness of this method for intelligent recommendation algorithms needs to be improved.

The above research applied graph attention to the recommender system and solved the problems of data sparsity and cold start to some extent. However, how to use graph attention networks to improve the real-time performance of recommender systems still needs to be studied. In this paper, the time factor is integrated into the graph attention network to generate a dynamic knowledge graph. Moreover, in order to reduce the impact of exposure bias and selection bias, additional relationships of the knowledge graph are defined.

2.3. RippleNet Application Status. The current typical method based on path and knowledge graph embedding is RippleNet [13]. Interest propagation is the important approach to implement RippleNet. Interest propagation can make full use of the user's historical data to obtain the preferences and then expand the user's interests outward along the relationship of the knowledge graph [45]. However, RippleNet does not consider the weight difference of the relationship between data. It is one kind of an undifferentiated transmission strategy of excellent seeds, which aggravates the impact of selection deviation and exposure deviation.

Luo et al. put forward the CN-RippleNet method [46], which combines the relevant knowledge of the complex network to calculate the influence of each node, and incorporates the influence into the original model. The model includes a user data processing module, recall layer module, and sorting layer module. However, the result of triple extraction of the improved method is inaccurate, and the node influence does not consider the relationship type. Shi fused two RippleNet models based on the knowledge graph to build a new recommendation model [47]. This model can discover the distribution of user interests and item features on the knowledge graph and the relationship between them. However, the attribute labels of each entity need to be explored. Luo et al. focused on the weight of entities and proposed a RippleNet model considering the influence of complex network nodes [48]. After constructing the complex network based on a knowledge graph, the maximum subnet model is established. The node influence in the graph network is calculated and added to RippleNet as a weight. Wang et al. proposed a multitask feature learning method based on RippleNet's knowledge graph enhancement in order to mine potential preferences from the knowledge graph [49].

This paper proposes a recommendation method based on a dynamic knowledge graph. First, multiplatform information is used to generate heterogeneous networks. User and item knowledge enhancement is achieved through multimodal nodes of knowledge graph. Based on the multimodal nodes, virtual relationships are introduced. Then, the multihead attention mechanism of the graph attention network is used to set the weight of each relationship in the knowledge graph. Finally, the improved RippleNet model is utilized to predict the user-item click-through rate, and a list of Top-N recommendation results with the highest probability value is given. The virtual nodes, virtual relationships, and advanced RippleNet mechanisms can effectively alleviate the problems of data sparsity and cold start and reduce the impact of data bias on the recommender system.

3. Adaptive Dynamic Knowledge Graph Recommender System

The overall framework of the proposed recommendation approach is shown in Figure 1.

The recommendation method includes three steps: (1) build a heterogeneous network based on multiplatform information and multimodal nodes and establish a basic knowledge graph; (2) integrate the time warehouse mechanism into GAT and use the graph attention network to extract the short-term preferences of users to obtain the real-time knowledge graph network; and (3) cluster users and items, optimize the RippleNet model based on excellent seed clusters, propagation blocking, and random seed mechanisms to predict click probability, and obtain a list of recommendations.

3.1. Building Heterogenous Network. In a recommendation environment, users' preferences on a platform can be supplemented and enhanced through users' behaviors on multiple platforms. Considering the complexity of user behavior and the multidimensional characteristics of user preferences, we first build a heterogeneous network which is shown in Figure 2. Several definitions related to heterogeneous networks are introduced as follows.

Definition 1. Multimodal nodes.

Multimodal nodes refer to multiple types of nodes in heterogeneous graphs, including users and item nodes. Besides the actual items that exist practically, item nodes also include virtual nodes such as topics, emotions, styles, and habits which are extracted from users' multi-platform information. Specifically, a virtual node is the user's preference or style in life, study, or work obtained by analyzing user's multi-platform behaviors. In Figure 2, virtual nodes are represented by dotted ellipses, such as work fanatic, anxiety tendency, and financial sector. Virtual nodes help discover the implicit preferences of users. Users and items in heterogeneous networks are represented by U and V, respectively.

$$\begin{aligned} U &= \{u_1, u_2, \cdots, u_n\}, \\ V &= \{v_1, v_2, \cdots, v_m\}, \end{aligned} \tag{1}$$

where U is a set of users, including n single users. V is a set of items, including m items.

Definition 2. Heterogeneous relationship.

The set of relationship types in heterogeneous networks is represented by R.

$$R = \{r_1, r_2, \cdots, r_l\}, \tag{2}$$

where $r_1, r_2, \cdots, r_l$ represents different types of relationships. Since there are multimodal nodes, the relationship between nodes contains many types. There are l types of relationships in set R, including multiple relationship types such as user's clicking on an item, user's interacting with another user, and item's relationship with other items. Among these l types of relationships, one part is the access or subordinate relationship between users and items based on explicit history records, which is called the inherent relationship. The other part is the relationship between user nodes and virtual nodes, which is called the virtual relationship. Virtual relationship is shown by the dotted lines in Figure 2. The relationships marked with question marks in Figure 2 refer to the ones that need to be predicted.

Definition 3. Relationship weight function.

RW denotes the relationship weight function between users. The weight function of the relationship r_i is represented as RW_{r_i},

The specific relationship weight of relationship r_i is calculated as follows.

$$\mathrm{RW}_{r_i} = \gamma \ln\left(\exp\left\{T_{r_i}\right\} + \exp\left\{F_{r_i}\right\} + \exp\left\{M_{r_i}\right\}\right), \tag{3}$$

where T_{r_i} represents the establishment time of the relationship r_i. F_{r_i} is the interaction frequency under the relationship r_i. M_{r_i} represents the number of mutual interaction or associated nodes of two nodes in relationship r_i. The longer the relationship between nodes is established, the greater the impact between nodes. For example, if two people have been good friends for many years, they usually share similar interests and hobbies. Therefore, they are easily influenced by each other. The higher the interaction frequency between nodes, the greater the influence exerted by each other. The more the number of jointly associated nodes of two nodes, the closer the relationship between nodes, and the greater the similarity weight of user social relations or items.

γ is the normalization coefficient, and it is used to reduce the deviation of the recommended results due to the large difference of the weight function. After constructing a heterogeneous network with weight values, the entities and relationships of the heterogeneous network are extracted, and the knowledge graph G is established to form the ternary relationship group (h, r, t), h, r, and t refers to head, relation, and tail, respectively. After constructing a heterogeneous network with weight values, the entities and relationships of the heterogeneous network are extracted, and the knowledge graph G is established to form the ternary relationship group (h, r, t), h, r, and t refers to head, relation, and tail, respectively.

3.2. GAT Integrated into the Time Warehouse Mechanism. In the recommender system, the user's behavior is usually affected by network interactions, network public opinions, and emergencies, and the relationship of nodes in heterogeneous graphs will change accordingly, which will lead to changes in short-term preferences, and short-term

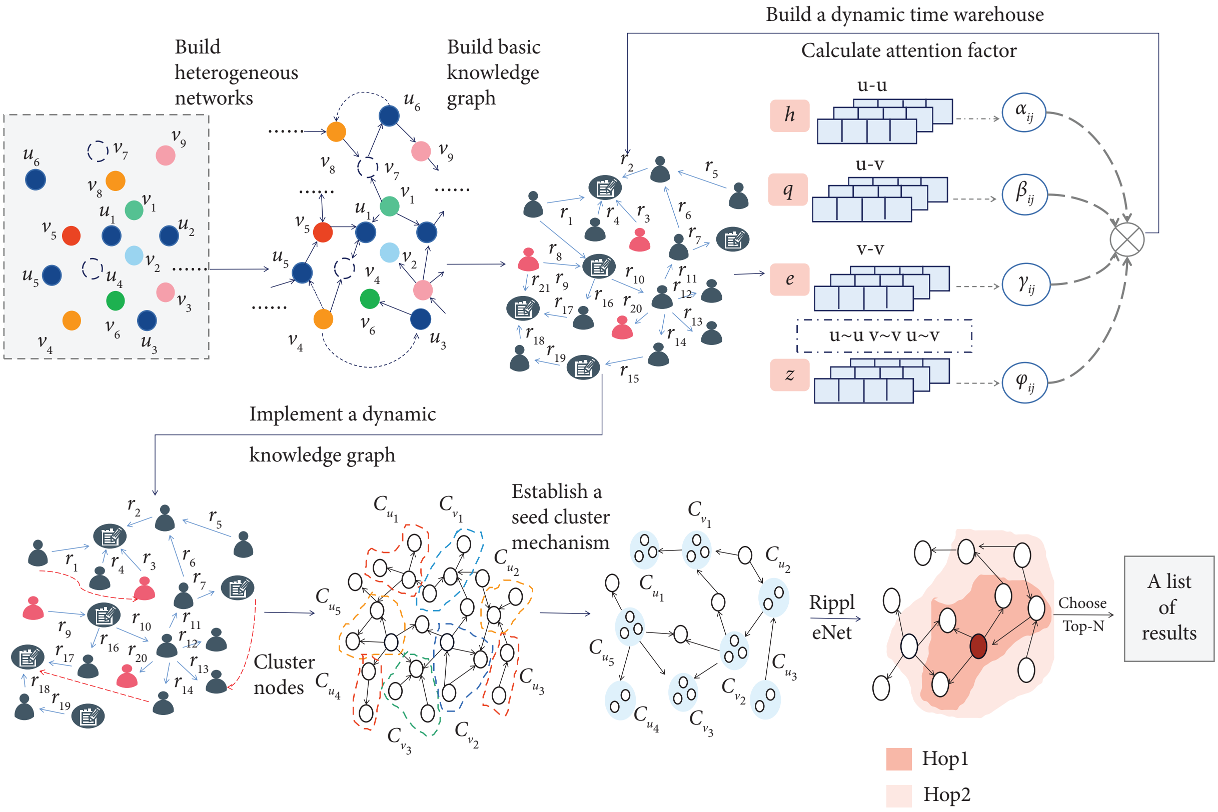
Build a dynamic time warehouse
Calculate attention factor
Build heterogeneous networks
Build basic knowledge graph
u-u
u-v
v-v
u~u v~v u~v
h
q
e
z
α_{ij}
β_{ij}
γ_{ij}
φ_{ij}
Implement a dynamic knowledge graph
Establish a seed cluster mechanism
Cluster nodes
Rippl eNet
Choose Top-N
A list of results
Hop1
Hop2

Figure 1: Overall architecture of the proposed HN-DKG.

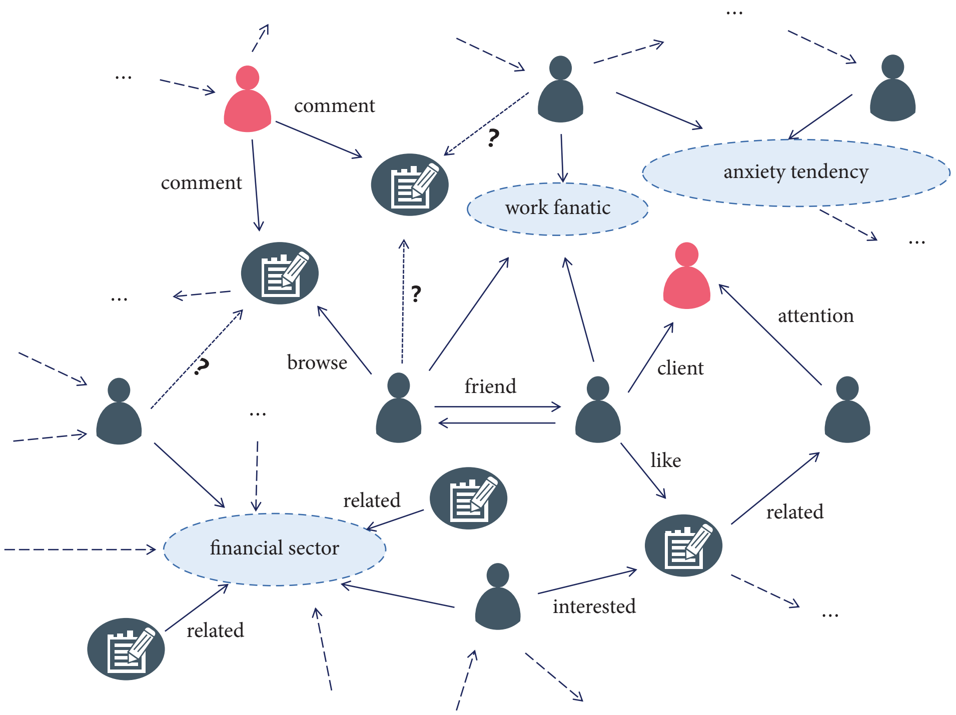

Figure 2: Multimodal heterogeneous network relationship diagram based on multiple platforms.

preference may also develop into long-term preference. GAT focuses on the neighbors of the target user, that is, the local structure of the graph. Hence, GAT can effectively extract users' short-term preferences. For this reason, we set a time warehouse and time warehouse trigger mechanism to facilitate GAT to extract users' short-term preferences.

Definition 4. Time warehouse.

A time warehouse is a time segment used to observe user behavior. TI_a is used to represent a time warehouse, and it can be represented as follows:

$$TI_a = [ti_a, ti_{a+1}]. \tag{4}$$

Usually, the time warehouse periodically extracts user preferences on a weekly basis. In the event of network public opinion and emergencies, we set the trigger function to reduce the space of the time warehouse, increase the number of warehouses, and increase the frequency of calculating user preference characteristics.

The sliding trigger function can be expressed as follows:

$$f(x) = \arctan\{\xi + \lg(\text{Poe} + \text{Eme}) + \tau_1 \lg(\text{Kes} + \text{Pug}) + \tau_2 \lg(\text{Fot} + \text{Ffi})\}, \tag{5}$$

where $f(x)$ is the trigger function, whose value is related to several behavior parameters, such as Poe-public opinion events, Eme-sudden/hot events, Kes-keyword search changes, Pug-purchase items, Fot-focus topics, and Ffi-frequent friend interactions. ξ is a constant and can be set empirically to adjust the trigger frequency of the time warehouse. Figure 3 shows the specific process of the time warehouse embedding layer to extract user features. It can trigger the creation of multiple time warehouses and can also perform feature extraction calculation of conventional time warehouses, which realizes the flexibility of time warehouse establishment, ensures system accuracy, and reduces calculation costs.

3.3. Multihead Attention Network of GAT.

Considering the complex relationship of multiple nodes, we use the multihead attention mechanism to extract the relationship of the dynamic knowledge graph, so as to make the attention weight more accurate and improve the accuracy of the recommendation results.

We focus on user-user, user-item, item-item, and additional relationships. A similar logical structure is used for function computation in these four relationships.

3.3.1. User-User GAT.

For the interaction between users, graph attention networks can be used to enhance the characteristics of user relations and mine implicit friends according to the path relationship of the knowledge graph. The potential characteristics between user u_j and other users are expressed as $h_j^{TI_a}$.

$$h_j^{TI_a} = \sigma\left(W \bullet \text{AF}_{u-u}\left(\left\{x_{ia}^{TI_a}, Ex_u \cup Im_u\right\}\right) + b\right), \tag{6}$$

where σ is the nonlinear activation function. AF_{u-u} is an aggregation function that integrates users' explicit friends and implicit friends. b is the neural network offset. W represents the neural network weight, which can be obtained by iterative training. Ex_u represents explicit friend feature representation. Im_u represents implicit friend feature representation. $x_{ia}^{TI_a}$ represents the interaction of the user with other users at time TI_a, such as chatting, following, and focusing. The attention coefficients of neighborhood users are represented as $\alpha_{ij}^{TI_a}$, which is calculated according to the latent features.

$$\alpha_{ij}^{TI_a} = \text{Softmax}\left(\overrightarrow{a}^T \bullet \sigma\left(W' \bullet \left[x_{ia}^{TI_a} \| h_j^{TI_a}\right] + b_1^k\right) + b_2^k\right). \tag{7}$$

Finally, the user's low dimensional vector feature representation is obtained. In order to improve the accuracy of its calculation, the multihead attention mechanism of the GAT model is used. The specific process of calculating $\overrightarrow{h}_j^{TI_a}$ is as follows.

$$\overrightarrow{h}_j^{TI_a} = \sigma\left(\frac{1}{K}\sum_{k=1}^{k}\sum_{j\in N_i}\left(\alpha_{ij}^{TI_a}\right)^k W^k \bullet x_{ia}^{TI_a} + b^k\right). \tag{8}$$

3.3.2. User-Item GAT.

The items that users have interacted with can be divided into two categories. One is direct user-item interaction, such as user evaluation, purchase, and collection, and the other one is an indirect interaction between users and virtual items based on the path in the heterogeneous graph. The representational feature $q_j^{TI_a}$ of user-item interaction can be expressed as follows.

$$q_j^{TI_a} = \sigma\left(W \bullet AF_{u-v}\left(\left\{s_{ia}^{TI_a}, Ex_v \cup Im_v\right\}\right) + b\right), \tag{9}$$

where σ is the nonlinear activation function. AF_{u-v} is an aggregation function that combines explicit interest items that users have directly interacted with and implicit items that users have indirectly interacted with through meta paths. b is the neural network offset. W represents the neural network weight, which can be obtained by iterative training. Ex_v indicates interesting items that users have interacted with in history. Im_v indicates an implicit item that users interact with indirectly through the Meta path. $s_{ia}^{TI_a}$ represents the user's interaction with other items at time TI_a. The attention coefficient of the neighborhood items, $\beta_{ij}^{TI_a}$, is calculated according to the latent features, and the normalized calculation is performed.

$$\beta_{ij}^{TI_a} = \text{Softmax}\left(\overrightarrow{a}^T \bullet \sigma\left(W' \bullet \left[s_{ia}^{TI_a} \| q_j^{TI_a}\right] + b_1^k\right) + b_2^k\right). \tag{10}$$

The output feature representation of user-item interaction is represented as follows:

$$\overrightarrow{q}_j^{TI_a} = \sigma\left(\frac{1}{K}\sum_{k=1}^{k}\sum_{j\in N_i}\left(\beta_{ij}^{TI_a}\right)^k W^k \bullet s_{ia}^{TI_a} + b^k\right). \tag{11}$$

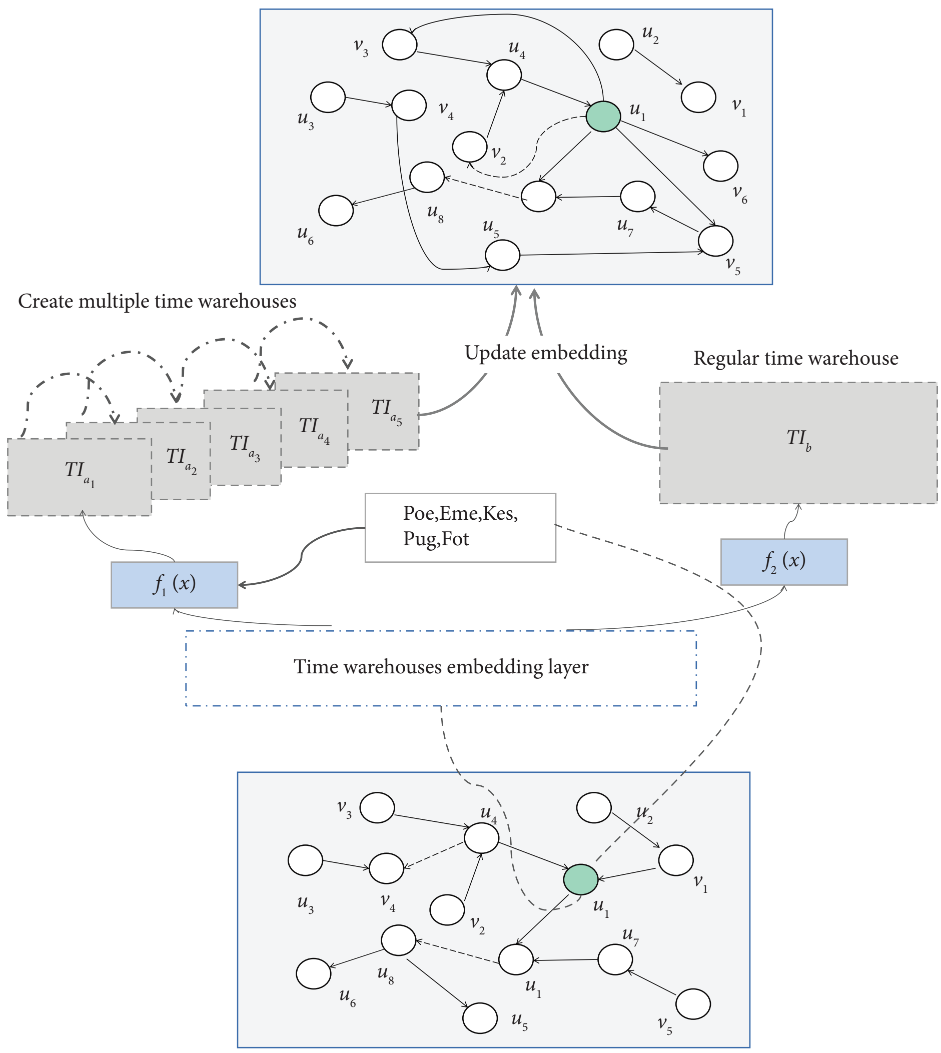

Figure 3: Embedding variable time warehouse into a multihead attention network layer.

3.3.3. Item-Item GAT. For the interaction information between items, we focus on the degree of association between historical interaction projects and neighborhood items, so as to provide users with better recommendations among the same type of items. The information between items includes direct information and indirect information. Direct information refers to the relationship that can be established between items through keywords or other attributes. Indirect information refers to the connection between items established through the user's social interaction, and also through the user's personality or style. The potential feature $e_j^{TI_a}$ between items can be calculated by the following formula. The potential characteristics between items are calculated by the following equation.

$$e_j^{TI_a} = \sigma\left(W \bullet AF_{v-v}\left(\left\{\rho_{ia}^{TI_a}, Di_v \cup In_v\right\}\right) + b\right), \tag{12}$$

where σ is the nonlinear activation function. AF_{v-v} is the aggregation function that fuses the information directly related to the target item and the information indirectly related to it. b is the neural network offset. W represents the neural network weight, which can be obtained by iterative training. Di_v refers to the item with information directly related to the target item. In_v refers to the item with indirect information related to the target item. $\rho_{ia}^{TI_a}$ represents the interaction embedding of the target item with other items at time TI_a.

The attention coefficient of the item, $\gamma_{ij}^{TI_a}$, is calculated according to the latent features, and the normalized calculation is performed.

$$\gamma_{ij}^{TI_a} = \mathrm{Softmax}\left(\overrightarrow{a}^T \bullet \sigma\left(W' \bullet \left[\rho_{ia}^{TI_a} \| e_j^{TI_a}\right] + b_1^k\right) + b_2^k\right). \tag{13}$$

The output feature representation of item-item is listed as follows.

$$\overrightarrow{e}_j^{TI_a} = \sigma\left(\frac{1}{K}\sum_{k=1}^{k}\sum_{j\in N_i}\left(\gamma_{ij}^{TI_a}\right)^k W^k \bullet \rho_{ia}^{TI_a} + b^k\right). \tag{14}$$

3.3.4. Additional Relationship GAT. In order to improve the diversity of recommendation results and reduce the impact of data bias on recommendation results, additional relationships are designed. There are two types of additional relationships. One is random recommendation to all users based on hot topics in the time warehouse, and the other is random recommendation to users that have no interaction with user's preferences. The nodes recommended to users in the additional relationship are called additional nodes.

Additional nodes are characterized by $z_j^{TI_a}$, which is represented as follows.

$$z_j^{TI_a} = \sigma\left(W \bullet \mathrm{AF}_{u\cdots v}\left(\left\{d_{ia}^{TI_a}, Vi_v\right\}\right) + b\right), \tag{15}$$

where σ is the nonlinear activation function. $\mathrm{AF}_{u\cdots v}$ is the aggregate function of users and additional nodes. b is the neural network offset. W represents the neural network weight, which can be obtained by iterative training. Vi_v represents the characteristics of additional nodes, and its number can be limited on the threshold according to the size of the dataset. $d_{ia}^{TI_a}$ represents the embedding of user and additional nodes at time TI_a. The attention coefficient of additional nodes, $\varphi_{ij}^{TI_a}$, is calculated according to the latent features, and the normalized calculation is performed.

$$\varphi_{ij}^{TI_a} = \mathrm{Softmax}\left(\overrightarrow{a}^T \bullet \sigma\left(W' \bullet \left[d_{ia}^{TI_a} \| z_j^{TI_a}\right] + b_1^k\right) + b_2^k\right). \tag{16}$$

The output characteristic of additional nodes is represented as follows.

$$\overrightarrow{z}_j^{TI_a} = \sigma\left(\frac{1}{K}\sum_{k=1}^{k}\sum_{j\in N_i}\left(\varphi_{ij}^{TI_a}\right)^k W^k \bullet d_{ia}^{TI_a} + b^k\right). \tag{17}$$

Figure 4 shows the structure of the attention network. It is divided into three layers, namely, the propagation layer, the time warehouse embedding layer, and the aggregation layer. The attention network carries out input propagation for attention calculation of four types of relationships which are user-user, user-item, item-item, and additional relationships. Through the calculation of multiple time slots, the dynamic attention weight is obtained. It can not only mine the potential preferences of users but also provide the possibility for the expansion and extension of user preferences. This will help improve the accuracy and diversity of recommendation results and improve user satisfaction.

3.4. Advanced RippleNet. After using GAT to extract users' short-term preference features, a real-time dynamic knowledge graph is obtained. Then, the RippleNet model is advanced to expand the knowledge graph information to complete the user's click prediction of the item. First, the weight coefficients generated by the graph attention network are used for quick clustering of multitype nodes. Each cluster is used as the seed cluster of RippleNet to propagate the knowledge graph. Then, in order to reduce the computational complexity, a propagation blocking mechanism is set up to determine the number of hops of propagation. For some island users or users with less historical access data, recommendations are made according to the weight reference function and random seed mechanism to alleviate the problem of data sparsity and obtain more diversified recommendations.

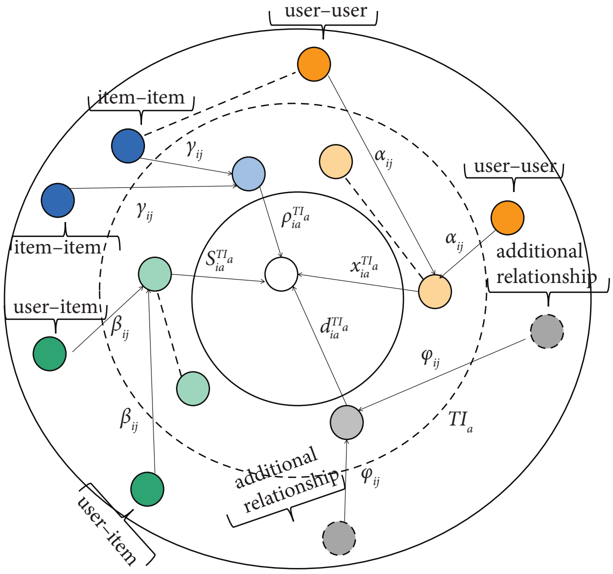

Figure 4: Attention structure diagram constructed from four types of node relationships.

Definition 5. Seed clustering.

According to the weight of the attention coefficient, users and items are respectively clustered by density-based clustering algorithm (DBSCAN) to generate node clusters. The seed clusters can not only ensure the compactness of node associations within the cluster but also ensure the diversity of nodes within the seed cluster. The optimality and diversity of seed clusters can ensure the effectiveness of RippleNet. The number of nodes contained in each cluster is related to the size of the dataset. The user cluster is represented as C_u. C_u^i is ith user cluster. $C_u^i = \{C_u^{i1}, C_u^{i2}, \cdots, C_u^{in}\}$. C_u^i includes n nodes. The item cluster is represented as C_v. C_v^i is ith item cluster. $C_v^i = \{C_v^{i1}, C_v^{i2}, \cdots, C_v^{im}\}$. C_v^i includes m nodes. User clusters interact directly or indirectly with item clusters. Specifically,

$$Cf_i = \nabla\left[\left(\log\sum_{a=m}^{n}\left(\alpha_{ij}^{TI_a} + \varsigma \bullet \beta_{ij}^{TI_a} + \epsilon \bullet \gamma_{ij}^{TI_a}\right)\right), \left(\exp\Omega \bullet \sum_{a=m}^{n}\varphi_{ij}^{TI_a}\right)\right]. \tag{18}$$

The threshold can be set according to needs or expert experience to classify user clusters and item clusters. The function increases the weight of the virtual relationship and attaches importance to the diversity of results. Among them, ς, ϵ, Ω is a normalized parameter setting, maintaining the order of magnitude of the parameter.

Definition 6. Propagation and blocking mechanisms.

To improve RippleNet, we set excellent seed, random seed, and propagation blocking mechanisms to guarantee personalized recommendation, reduce the impact of exposure deviation, and improve recommendation efficiency, respectively.

The interaction matrix between user cluster and item cluster is expressed as Y.

$$Y = \left\{y_{C_u C_v} | C_u \subset U, C_v \subset V\right\}, \tag{19}$$

where $y_{C_u C_v}$ represents the interaction coefficient between user clusters and item clusters. $y_{C_u C_v}$ has three values.

(i) If $y_{C_u C_v} = 1$, it means the user cluster has direct interaction with the item cluster or indirect interaction along the meta path of the graph data. Then, the item cluster is called the excellent seed cluster set for the user cluster. The propagation of excellent seeds is indicated by blue arrows in Figure 5.

(ii) If $y_{C_u C_v} = 0$, it means that there is no interactive information between the user cluster and item cluster. These item clusters can be used as candidate seed sets of user clusters to establish a random propagation relationship between users and items. The propagation of random seeds is indicated by green dotted arrows in Figure 5.

(iii) If $y_{C_u C_v} = -1$, it indicates that the relationship between the user cluster and the item cluster is an inhibition relationship. The item cluster is set as an inhibition cluster to block the propagation between the items and the users. The red cross is used to represent the propagation interruption in Figure 5.

Specifically, the size of the interaction matrix is calculated according to the weight function calculated above.

$$qa_x = \frac{\sum_{i=1,j=1}^{i=L,j=L}\left(\sum_{a=m}^{n}\alpha_{ij}^{TI_a}\|\sum_{a=m}^{n}\beta_{ij}^{TI_a}\|\sum_{a=m}^{n}\lambda_{ij}^{TI_a}\|\Omega\sum_{a=m}^{n}\phi_{ij}^{TI_a}\right)}{\vartheta\sum_{i=1}^{L}N_{C_u C_v}^{TI_i}}. \tag{20}$$

The threshold function is set as $\text{crit} = \sqrt{1/qa\,[\ln(\text{qa})/\ln 2]^3}$, $\text{eta} = [\sum_{x=1}^{N}|qa_x|^2 - N']/N'$. If $\text{eta} < \text{crit}$ and $y_{C_u C_v} = 1$, the prediction probability will be calculated. If $\text{eta} > \text{crit}$ and $y_{C_u C_v} = -1$, the propagation blocking mechanism is enabled, and the prediction probability is not calculated. N' is a fixed value, so that the eta, crit function can be compared in the same order of magnitude. If $qa_x = 0$, it means that there is no interaction between the two nodes. A virtual relationship is randomly established, and the prediction probability is calculated to find the user's potential preference.

Define 7. Node cluster set.

The set of entity node clusters is represented by $\varepsilon_{C_u}^{k}$.

$$\varepsilon_{C_u}^{k} = \left\{t\,|\,(h,r,t) \in G\,\text{and}\,h \in \varepsilon_{C_u}^{k-1}\right\} k = 1,2,\cdots\cdots H, \tag{21}$$

where the above formula represents the set of user clusters and item clusters associated with user clusters after k jumps. Wherein, $\varepsilon_{C_u}^{0} = \left\{\varepsilon\,|\,y_{C_u C_v} = 1, y_{C_u C_v} = 0\right\}$, it represents the collection of excellent seed clusters and random seed clusters.

Define 8. Ripple sets.

The ripple set in RippleNet is set as $S_{C_u}^{k}$.

$$S_{C_u}^{k} = \left\{(h,r,t)|\,(h,r,t) \in G\,\text{and}\,h \in \varepsilon_{C_u}^{k-1}\right\} k = 1,2,\cdots\cdots H. \tag{22}$$

The potential interest of user clusters to item clusters is expanding, but with the expansion of the scope of propagation, the intensity of preference transmission is also gradually weakening. The framework of RippleNet is also shown in Figure 5.

The user marked with red color in a cluster is the targeted user. The interaction of other users in the cluster and the item cluster of their own interaction can be used as the seed cluster for calculating the prediction probability. Virtual relationships are randomly established for island users to enrich user information. At the same time, the inhibition items will be blocked and no information will be transmitted from these items.

By comparing the feature C_v of the item cluster with the head node h_i and relation r_i of the triplet (h_i, r_i, t_i), the association probability of each triplet in the ripple set $S_{C_u}^{1}$ can be obtained. The formula of P_i is listed as follows.

$$\begin{aligned} P_i &= \text{softmax}\left(C_v^T R_i h_i\right) \\ &= \frac{\exp\left(C_v^T R_i h_i\right)}{\sum_{(h,r,t)\in S_{C_u}^{l}}\left(C_v^T R h\right)}, \end{aligned} \tag{23}$$

where R_i and h_i are the features of the relation r_i and head node h_i, respectively. Then, the weighted sum of the tail nodes in $S_{C_u}^{1}$ is calculated, and the weight is the correlation probability calculated by (24), and the vector $O_{C_u}^{1}$ is obtained.

$$O_{C_u}^{1} = \sum_{(h_i,r_i,t_i)\in S_{C_u}^{l}} p_i t_i, \tag{24}$$

where t_i is the feature of the tail node t_i, and the vector $O_{C_u}^{1}$ represents the first-order response of the user cluster C_u in the seed set of the knowledge graph to the item cluster. The corresponding expansion is carried out, the multiorder response is calculated, and the summation is performed to obtain C_u, which is the response after integrating all orders.

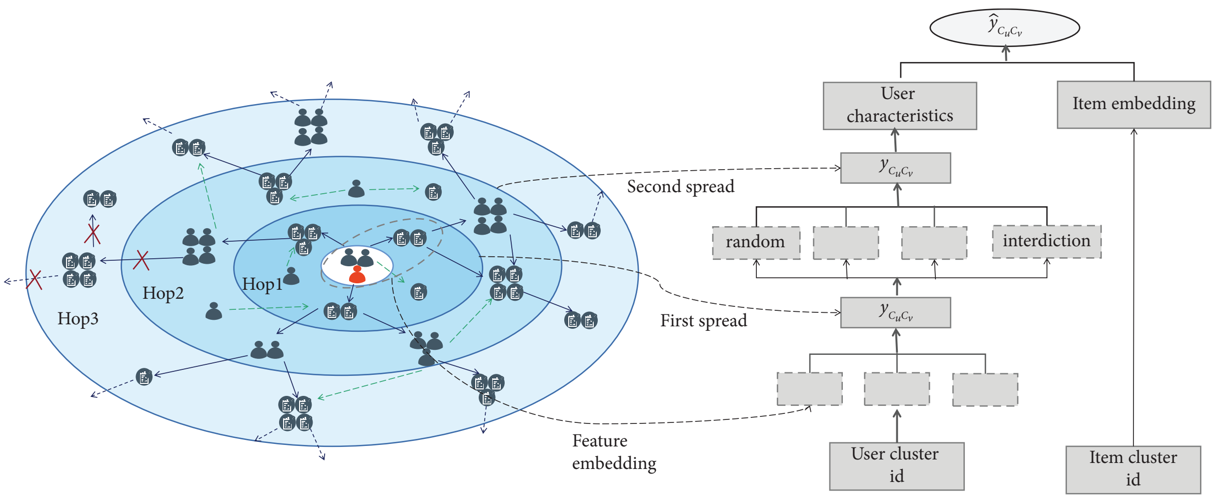

Figure 5: Framework of advanced RippleNet. It incorporates a propagation blocking mechanism, cluster of users and items, and random propagation relationship.

$$C_u = O^1_{C_u} + O^2_{C_u} + \cdots O^H_{C_u}. \tag{25}$$

Finally, combining user clusters and item clusters, the predicted user clicking probability is output. The calculation formula is as follows.

$$\begin{aligned}\hat{y}_{C_uC_v} &= \sigma\left(C_u^T C_v\right)\\ &= \frac{1}{\left[1 + \exp\left(C_u^T C_v\right)\right]}.\end{aligned} \tag{26}$$

3.5. Algorithm Description. The key steps of HN-DKG include how to extract user short-term preference and how to predict the user click probability. Hence, we introduce two algorithms, which are the feature extraction algorithm of GAT and probability prediction algorithm of RippleNet. At the same time, Algorithm 1 is given to illustrate how the above methods cooperate. In which, $\mathrm{acc}_{\mathrm{th}}$ and $\mathrm{loss}_{\mathrm{th}}$ are the thresholds set for prediction accuracy and cross entropy loss, respectively. Their values can be determined based on practical needs for model training or expert experience.

3.5.1. GAT Algorithm. Algorithm 2 is mainly composed of three parts. The first part (lines 1–4) constructs a time warehouse to preprocess data and extract user and item features. The second part (lines 5–7) calculates the attention coefficient and low dimensional representation of user and item characteristics. The third part (lines 8–12) adjusts the heterogeneous network according to the weight of the attention coefficient to achieve dynamic network representation and output potential features.

3.5.2. RippleNet Algorithm Description. Algorithm 3 is mainly composed of three parts. In the first part (lines 1-2), data preprocessing is used to extract user clusters and item clusters as the seeds of the corrugated network model. The second part (lines 3–9) calculates the multihop vector of the corrugated network and processes and calculates the triplet relationship. The third part (lines 10–13) calculates the multiorder response and prediction probability and finally gives a list of recommendation results. In Algorithm 3, P_{th} is the threshold set for triple association probability and K_{set} is the threshold for the number of iterations to calculate the optimal P_i. Their values can be determined based on practical needs for model training or expert experience.

4. Experimental Analysis

This section introduces the experimental settings, experimental datasets, the comparison algorithms, and the parameters settings. Then, the experimental results are analyzed.

4.1. Experimental Dataset. In this study, the datasets of MovieLens-1M movies and Book Crossing books are applied on the performance test. Table 1 shows the details of the two datasets.

MovieLens-1M (https://grouplens.org/datasets/movielens/1m/) is a standard dataset widely used in the field of movie recommendation field, which includes 1000209 ratings from 6036 users on 2445 items. Each score is a positive integer between 1 and 5. Book-Crossing (http://www2.informatik.uni-freiburg.de/~cziegler/BX/) is a standard dataset widely used in the field of book recommendation. This dataset contains 1149780 ratings from 70679 users on 24915 items. Each score is a positive integer between 1 and 10. After removing fuzzy relation, data screening, and data deduplication, the knowledge map in the dataset is extracted by similarity measurement [50] and sampling [51]. The knowledge graph corresponding to the dataset MovieLens-1M contains 182011 entities, 12 different relationships, and 1241995 pieces of knowledge, while the dataset Book Crossing contains 113487 entities, 80 different relationships, and 6420520 pieces of knowledge.

```
    Input: Multiplatform data d$t$
    Output: Recommended Result List Top-10
(1) Preprocessing d$t$, building heterogeneous networks HN
(2) Extract ternary relationship $(h, r, t)$ from HN and establish basic knowledge graph $G$
(3) Calculate GAT to extract users and items' short-term preference features $h^{TI_a}, q^{TI_a}, e^{TI_a}, z^{TI_a}$ within the time warehouse $TI_a$
(4) Update the basic knowledge graph $G$
(5) User clustering $C_{u_j}$ and item clustering $C_{v_j}$
(6) Use improved RippleNet to calculate prediction accuracy auc and cross entropy loss loss
(7) If auc is greater than acc$_{\mathrm{th}}$ or loss is less than loss$_{\mathrm{th}}$ then jump to 8, otherwise, jump to 3
(8) Output recommended result list Top-10
```

Algorithm 1: HN-DKG.

```
     Input: user and item feature $h_j^{TI_a}, q_j^{TI_a}, e_j^{TI_a}, z_j^{TI_a}$
     Output: low-dimensional representation $\overrightarrow{h}_j^{TI_a}$, $\overrightarrow{q}_j^{TI_a}$, $\overrightarrow{e}_j^{TI_a}$, and $\overrightarrow{z}_j^{TI_a}$ of latent features of users and items, and weights of attention coefficients $\alpha_{ij}^{TI_a}$, $\beta_{ij}^{TI_a}$, $\gamma_{ij}^{TI_a}$, and $\varphi_{ij}^{TI_a}$
 (1) $a = 1, i = 1, j = 1$
 (2) Build time warehouse $TI_a = [ti_a, ti_{a+1}]$
 (3) Extract feature representations of users and items in time warehouse $TI_a$
 (4) Data preprocessing
 (5) Calculate the initial eigenvectors $h_j^{TI_a}$, $q_j^{TI_a}$, $e_j^{TI_a}$, and $z_j^{TI_a}$
 (6) Calculate the attention coefficients $\alpha_{ij}^{TI_a}$, $\beta_{ij}^{TI_a}$, $\gamma_{ij}^{TI_a}$, and $\varphi_{ij}^{TI_a}$
 (7) Calculate the low-dimensional feature vectors $\overrightarrow{h}_j^{TI_a}$, $\overrightarrow{q}_j^{TI_a}$, $\overrightarrow{e}_j^{TI_a}$, and $\overrightarrow{z}_j^{TI_a}$
 (8) If $\alpha_{ij}^{TI_a}, \beta_{ij}^{TI_a}, \gamma_{ij}^{TI_a}, \varphi_{ij}^{TI_a} \in (0, m)$, then delete this type of relationship, else embed heterogeneous network
 (9) $i \leftarrow i + 1, j \leftarrow j + 1$.
(10) Jump to step 5.
(11) Output retention attention coefficient.
(12) Output low-dimensional feature vector representation.
```

Algorithm 2: Graph attention network algorithm.

```
     Input: interaction matrix $Y$, knowledge graph $G$
     Output: prediction function $\hat{y}_{C_u C_v} = \sigma(C_u^T C_v)$, recommendation result list $L$
 (1) Initialize all parameters
 (2) According to the weight value of the graph structure, collect the user cluster $C_{u_j}$ and the item cluster $C_{v_j}$
 (3) Calculate $Y$ for each user cluster according to the interaction matrix $\varepsilon_{C_u}^H$
 (4) $k = 1$
 (5) Calculate the triple association probability $P_i$ of the ripple set $S_{C_u}^k$
 (6) When $(h_i, r_i, t_i) \in S_{C_u}^k$
 (7) Calculate the weighted sum $O_{C_u}^k$ of the tail node
 (8) $k \leftarrow k + 1$
 (9) If $P_i > P_{\mathrm{th}}$ or $K > K_{\mathrm{set}}$, jump to 10, otherwise, jump to step 5
(10) Calculate the multiorder response vector and $C_u$
(11) Calculate the predicted click probability $\hat{y}_{C_u C_v}$
(12) Sort the probability values to give Top-N item clusters
(13) Output probability and list of recommended results
```

Algorithm 3: RippleNet algorithm.

Table 1: Statistics of two datasets.

Dataset	#Users	#Items	#Entities	#Relations	#Triplets	Density (%)
MovieLens-1M	6036	2445	182011	12	1241995	4.47
Book-Crossing	70679	24915	113487	80	6420520	0.005

To test the performance of recommendation approaches on multiplatform, we construct a new dataset named MB-data which is based on heterogeneous platforms. The premise of designing this dataset is to assume that users who are interested in a certain field also have the same interest in other fields. The theoretical basis for this hypothesis is the user interest transfer theory, which states that there is a connection between the user's interest preferences in the source and target domains [52, 53].

The first construction method is to extract highly matched users as the same user in the MovieLens-1M and Book-Crossing datasets according to the distribution of user scores and the similarity of the characteristics of the item categories that users like and dislike and construct the interactive information of multi-platform to generate multi-platform datasets. Another construction method is to collect the interactive data of users on platforms such as Weibo, Taobao, and CSDN, such as purchasing, browsing, and evaluating, rating based on frequency and time, and establish a relational dataset to obtain richer multi-platform data.

HN-DKG uses movies, books, and MB data datasets. To reflect multiplatform data, we integrate user nodes in the movie and book datasets and then establish relationships between users and movie and book nodes. This forms a multimodal node dataset.

Further processing was done on the dataset to create a dataset suitable for our experiment. To ensure the connectivity of heterogeneous networks, we remove users who does not have multi-platform behaviors. The data were randomly divided into a training set (80%) and a testing set (20%) in an inductive way.

4.2. Contrast Experiment

4.2.1. Comparison Method. The following approaches are used as baselines for comparative experiments, as shown in Table 2.

(i) EHCF [54] (efficient heterogeneous collaborative filtering) can model fine-grained user item relationships.

(ii) CKE [55] (collaborative knowledge base embedding) uses heterogeneous network embedding and deep learning embedding methods to automatically extract semantic representation from structural knowledge, text knowledge, and visual knowledge in the knowledge base.

(iii) RippleNet [13] expresses user preferences through a large number of entities related to user click history. In this method, the knowledge graph representation method used is TransE [57].

(iv) KGAT [42] (knowledge graph attention network) explicitly models high-order connections in KG in an end-to-end manner.

(v) NFM [56] (neural factorization machine) proposed a new model neural factorization machine for prediction under sparse sets.

4.2.2. Evaluation Matrix. These recommendation methods aim to give the Top-N recommendations for users. To evaluate the performance of these recommendation methods, the precision, recall, and $F1$ score commonly are used as evaluation criteria.

The accuracy rate represents the number of items in the recommendation list for which users have had positive feedback as a percentage of the total number of items in the recommendation list. The formula of the accuracy rate is as follows.

$$\text{Precision} = \frac{\sum_{u\in U}|\text{TP}(u)|}{\sum_{u\in U}|\text{TP}(u) + \text{FN}(u)|}, \tag{27}$$

where U represents a set of users and u is a specific user in the set U.

Recall describes the proportion of the number of items in the recommended list that users have had positive feedback to the number of items in the test set. It is calculated as follows.

$$\text{Recall} = \frac{\sum_{u\in U}|\text{TP}(u)p|}{\sum_{u\in U}|\text{TP}(u) + \text{FN}(u)|}, \tag{28}$$

where $\text{TP}(u)$ and $\text{FN}(u)$ denote the goods that are predicted to be actually liked and disliked by user u among the goods that user u is interested in, respectively.

The precision rate is for the prediction result, which indicates how many of the samples predicted to be positive are actually positive samples. Recall, on the other hand, is for the original sample, and it indicates how many of the positive examples in the sample were predicted correctly.

$F1$ score combines the accuracy rate and recall rate to measure the effect of recommendation. The higher the value of $F1$ score, the better the effect of recommendation. The formula is as follows.

$$F1 - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}. \tag{29}$$

This parameter is used to compare and discuss the similar models and analyze the advantages and disadvantages of the proposed method and baseline methods.

Normalized discounted cumulative gain (NDCG) is an evaluation index that takes into account the return list to evaluate the accuracy of the list. The value is in the range of (0, 1). The larger the value, the better the recommendation effect. AUC measures the probability that a model will predict a positive sample as a positive example rather than a negative sample as a positive example. The formula is as follows.

$$\begin{aligned} \text{NDCG} &= \frac{\text{DCG}}{\text{IDCG}}, \\ \text{AUC} &= \frac{\left[\sum_{ins_i\in\text{positive}} \text{ran}k_{ins_i} - M * (M+1)/2\right]}{M} * N, \end{aligned} \tag{30}$$

where DCG refers to discounted cumulative gain, IDCG refers to optimal DCG, NDCG is used to evaluate the accuracy of ranking, $\text{ran}k_{ins_i}$ represents the sequence number

Table 2: Baseline methods used in the experiment.

Method	Proposer	Details
EHCF	Chen et al. [54]	It can efficiently learn model parameters from the entire heterogeneous data (including all unlabeled data), with low time complexity
CKE	Zhang et al. [55]	Combine collaborative filtering and knowledge base embedded components into a unified framework and learn different representations together
RippleNet	Wang et al. [13]	The click through rate is estimated through the combination of user preferences and item presentation
KGAT	Wang et al. [42]	Recursively propagate embedding from a node's neighbors (which can be users, items, or attributes) to optimize the embedding of nodes and use the attention mechanism to distinguish the importance of neighbors
NFM	He and Chua [56]	It combines the linearity of FM in the second-order feature interaction modelling and the nonlinearity of the neural network in the higher order feature interaction modelling

of samples with the ith smallest probability score. M and N represent the number of positive and negative samples, respectively. AUC is used to evaluate the ranking quality of predictions.

4.3. Experiment and Result

4.3.1. Experimental Environment. The experiments in this study were conducted on a desktop computer with 8 GB of RAM, an Intel Core i5-8250 CPU, and the software was Python 3.9. The other details of the experiment environment setting are shown in Table 3.

4.3.2. Setting Hyperparameters. The hyperparameters in the recommended approaches are determined by experiments, that is, when other parameters are determined, we change the values of the parameters needed to be determined and select the parameters that achieve the best experimental results. We take the number of heads of multihead attention as an example to illustrate the process.

The number of heads of multihead attention determines the number of times the model calculates. In general, the more heads the model has, the more accurate attention is extracted, but more heads will also increase the calculation cost and the possibility of overfitting. Therefore, in order to maximize the extraction of user features and minimize the calculation time of the model, it is crucial to select an appropriate number of heads. We test the performance of the number of multiple heads of HN-DKG in the case of $K = 2, 4, 6, 8, 10$.

Figure 6 shows the performance comparison of HN-DKG when using different K values on MovieLens-1Mdataset, where RMSE is the root mean square error and MAE is the mean absolute error. Figure 7 shows the precision of recommendations under different K values. Figure 6 and 7 indicate that the proposed method achieves lower RMSE and higher precision when K is 4. In addition, we compared Flops under different K values [58]. When $K = 4$, the proposed method only accounted for 0.14% of the computational cost in creating and applying attention matrices, which is only 0.05% higher than the Flops at the suboptimal $K = 2$. The experiment indicates that the proposed method can achieve a balance of precision, RMSE, and computational complexity at a K of 4 on the experimental data scale.

4.3.3. Performance Comparison. These recommendation models are tested on the MovieLens-1M dataset. 70% of the data are used for training and 30% are used for verification. The data set is split in an inductive way, that is, the data in the test set will not appear in the training set.

The selected evaluation indexes are precision, recall, $F1$ score, NDCG@10, and AUC. The experimental results are shown in Table 4. These methods are also tested on the Amazon Book dataset. The comparative experimental results are shown in Table 5.

In order to test the performance of the proposed HN-DKG algorithm under extremely sparse data, based on the Amazon Book dataset, we delete 2/3 of the scoring data to increase the sparsity of the data. RippleNet and KGAT, which performed better in the above experiments, are selected to compare and discuss the performance of the proposed model HN-DKG.

Figure 8 shows the performance comparison of three models under extremely sparse data. The model is iterated on MovieLens-1M dataset to obtain more accurate weights. Among these three indicators, HN-DKG is better than the other two methods.

The effects of training iterations on loss function and accuracy (acc) are compared through experiments. Figure 9 shows the change of loss function and accuracy rate with the increase of iterations. Both parameters are within the range of (0, 1), and the higher the value of loss, the less loss the model has and the better its functionality. The larger the acc value, the higher the accuracy and performance of the model.

4.4. Ablation Experiment. In order to evaluate the importance of the main innovations of the proposed algorithm, we designed ablation experiments. Specifically, the virtual node, the time warehouse mechanism, and the user clustering factor are considered in the ablation experiment. Also, we conducted the experimental results of ablation studies in Table 6.

Table 3: Experimental setting.

Parameter	TensorFlow GPU	NumPy	Pandas	Sklearn	Operating system	Display adapter
Version or model	1.14.5	1.18.5	1.1.3	0.23.2	Windows 11 64-bit	Nvidia RTX 2070 GPU

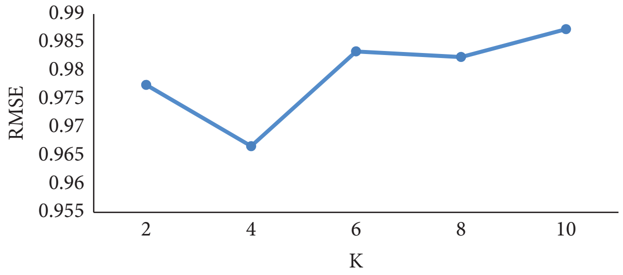

Figure 6: RMSE under different attention head K values.

In order to test the effect of virtual nodes on the effectiveness of these approaches, virtual nodes are removed and an algorithm HN-DKG-drop1 is formed. MovieLens-1M with a small amount of data is selected, and the improved HN-DKG model is compared with HN-DKG-drop1 to test the impact of virtual nodes on the recommendations. Besides the results of HN-DKG, the experimental results of the HN-DKG-drop1 model are shown in Table 6.

In order to test the effect of the time warehouse mechanism on the system effectiveness, the time warehouse is eliminated and the algorithm HN-DKG-drop2 is formed. MovieLens-1M, which has a small amount of data, is selected for this part of the experiment. The improved HN-DKG model is compared with HN-DKG-drop2 to test the impact of the time warehouse mechanism on the recommendation results. Besides the results of HN-DKG, the experimental results of the HN-DKG-drop2 model are shown in Table 6.

In order to test the effect of clustering on the recommender system, clustering factors are eliminated and an algorithm HN-DKG-drop3 is formed. In this part of the experiment, MovieLens-1M with a small amount of data is selected. The improved HN-DKG model is compared with the HN-DKG-drop3 to test the impact of clustering on the recommended results. In the experiment, for data preprocessing, a density-based clustering algorithm can be used to classify the data according to the reference factors of weights. Besides the results of HN-DKG, the experimental results of HN-DKG-drop3 model are shown in Table 6.

4.5. Analysis and Conclusion. Compared with the baseline model, the HN-DKG algorithm in this article has achieved the best overall performance among all evaluation indicators. Specifically, HN-DKG is 18%, 9%, and 2% higher than KGAT on F1, NDCG@10, and AUC and 20%, 2%, and 0.9% higher than RippleNet on MovieLens-1M dataset. HN-DKG is 12%, 3%, and 2.5% higher than NFM on F1, NDCG@10, and AUC and 0.8%, 2.3%, and 0.35% higher than RippleNet, respectively. This indicates that the proposed model can better mine user preferences, provide more accurate recommendation lists, and have real-time performance.

4.5.1. Experimental Results. When determining the number of heads K for multihead attention, comparing the RMSE and MAE parameters, it can be seen from Figure 6 that when $K = 4$, the performance is best, and the calculation time is relatively short, resulting in lower computational costs [59]. The performance trend of the model is to first reach the optimal point and then gradually decrease, indicating that increasing the number of attention heads can better mine user features and improve model performance. However, when there are too many heads, the performance of the model will decrease due to overfitting, and the calculation time will be longer, resulting in unsatisfactory model performance. Therefore, a K value of 4 was determined for subsequent experiments.

From Table 4, it can be seen that the HN-DKGT proposed in this article has higher accuracy than other algorithms, but slightly lower recall than some models. If we pay attention to the experimental results comprehensively, RippleNet model has a relatively high recall rate and slightly lower accuracy than other models. This also confirms some empirical results of the recommendation model, that is, accuracy and recall cannot be achieved simultaneously. Therefore, finding a balance between the two is the best way to optimize the model. The model proposed in this article achieves a high balance in the ratio $F1$ of accuracy and recall, verifying the effectiveness of the proposed method and also improving a certain degree of accuracy. According to Table 5, it can be seen that the model proposed in this article has the highest recall rate in the Amazon Book dataset, but the advantage of accuracy is not obvious enough. This may be because the dataset is relatively large and the improvement effect is not significant enough, but there is still a certain advantage in the numerical value of $F1$, achieving a relatively optimal balance. By analyzing the NDCG@10 and AUC in both Tables 4 and 5, we can conclude that the performance of HN-DKG is slightly higher than that of other models, which further proves the effectiveness of the proposed method.

According to Figure 8, in the MB-data dataset, the accuracy of the model proposed in this paper is 0.05% higher than that of RippleNet algorithm with good performance and 0.1% higher than that of EHCF algorithm with similar performance. At the same time, it performs best in similar algorithms on $F1$. It shows that the proposed method can achieve more accurate recommendation results by extracting effective auxiliary information for user/item completion and knowledge enhancement in multi-platform and interactive information data. It can be seen that the HN-DKG has slight advantages in accuracy and recall, and it achieves a better balance in $F1$. It is proved that the proposed method is effective in dealing with cold start and sparse data problems.

It can be seen from Figure 9 that with the increase of training iterations, the loss function gradually decreases and the accuracy rate continuously improves. The loss function

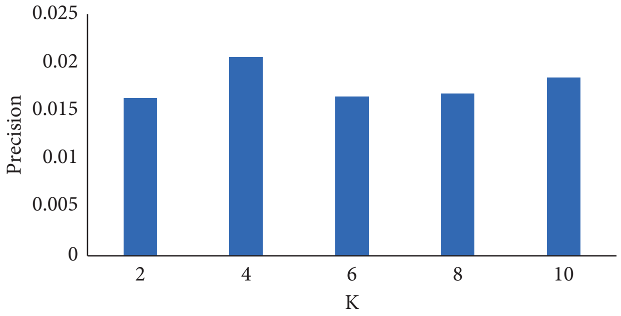

Figure 7: Precision under different attention head *K*.

Table 4: Experimental results on MovieLens-1M dataset.

Model	Precision	Recall	*F*1 score	NDCG@10	AUC
HN-DKG	0.02086	0.05813	0.03070	0.03161	0.65341
EHCF	0.01789	0.04895	0.02620	0.03013	0.63282
CKE	0.01648	0.05846	0.02572	0.02942	0.63488
RippleNet	0.01565	0.06875	0.02550	0.03075	0.64353
KGAT	0.01604	0.06584	0.02580	0.02894	0.63914
NFM	0.01495	0.06356	0.02420	0.02798	0.63876

Table 5: Experimental results on Amazon Book dataset.

Model	Precision	Recall	*F*1 score	NDCG@10	AUC
HN-DKG	0.01519	0.15412	0.02761	0.02865	0.54392
EHCF	0.01451	0.15284	0.02650	0.02761	0.53297
CKE	0.01354	0.14324	0.02475	0.02793	0.53657
RippleNet	0.01508	0.14800	0.02738	0.02801	0.54204
KGAT	0.01491	0.14146	0.02698	0.02742	0.53108
NFM	0.01350	0.13786	0.02459	0.02768	0.53017

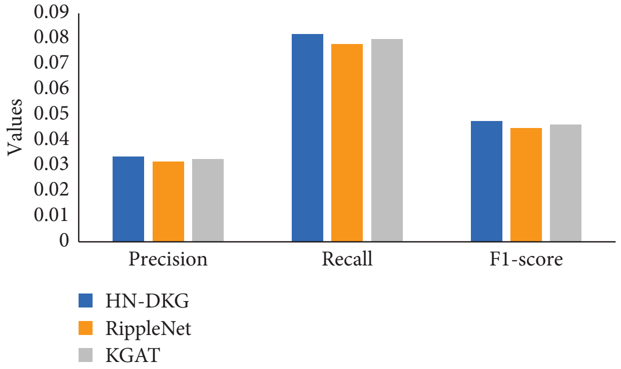

Figure 8: Parameter comparison among HN-DKG, RippleNet, and KGAT with sparse data.

of training data is generally lower than that of prediction data, and the accuracy of training data is generally higher than that of prediction data. However, when the training iterations are 100, both of the loss and acc are abnormal. It may be that due to the low number of training iterations, the calculation of the weight function file is biased, so the recommendation model still needs to increase the number of training iterations to get a stable and accurate weight function file.

4.5.2. Diversity Analysis. In order to evaluate the diversity of recommendation results, the attribute distribution of recommendation resources is analyzed first, and then, the long tail resources with low matching degree are recommended.

The diversity measurement method in the Top-N strategy is introduced to evaluate the diversity of recommendation results [60], that is, the diversity is calculated according to whether the attribute distribution of resources is balanced. Considering that the diversity function based on item characteristics can be seen as a supplement to the similarity measurement [61], we take the decentralized distribution of attributes as the diversity evaluation index. The calculation formula of diversity is as follows:

$$\mathrm{DI}(R) = \frac{1}{[|R| * (|R| - 1)] * \sum_{i\in R}\sum_{j\in R, j\neq i}\mathrm{div}\left(l_i, l_j\right)}, \tag{31}$$

where R is a set of resources recommended for user U, the number of R is recorded as $|R|$, and $\mathrm{div}\,(l_i, l_j)$ is a supplement to $\mathrm{sim}\,(l_i, l_j)$. The similarity calculation formula adopts cosine similarity algorithm. In general, improving diversity is at the cost of reducing the accuracy of recommendations. We will compare the diversity and accuracy of the recommended results.

Figure 10 shows the relationship between accuracy and diversity of recommendation results. The curve trends of EHCF, CKE, KGAT, and NFM in Figure 10 show that with the improvement of accuracy, diversity shows a downward trend. Compared with RippleNet, the diversity of HN-DKG declined slowly and remained above 0.40. Therefore, when the accuracy is improved, the diversity loss of HN-DKG is not very significant. The bottom-up recommendation strategy helps to improve the diversity of recommendation results. At the same time, it ensures the accuracy of recommendation results by enhancing the relationship between users' multi-platform data.

Through the above experiments, it can be known that the applications of multihead attention-based GAT model and the advanced RippleNet model are effective for the improvement of the recommender system. Due to the size of the dataset and the training degree, the weight files obtained are different, which has a certain impact on the accuracy and recall rate. However, this model has a certain advantage in the accuracy and recall ratio *F*1. The balance calculation between the two indexes is relatively stable, which improves the overall performance of the recommendation approach.

4.5.3. Analysis of Ablation Experiments. As shown in Table 6, compared with the original algorithm HN-DKG, the evaluation metrics of the training and verification model of HN-DKG-drop1 have decreased, with an average decrease of 2.57% on accuracy and 5.49 on AUC. It is proved that it is useful to add the virtual node factor into the recommender system to improve the system performance. Therefore, the original intention and the implementation of HN-DKG are effective.

Similarly, as shown in Table 6, compared with the original algorithm HN-DKG, it is found that the

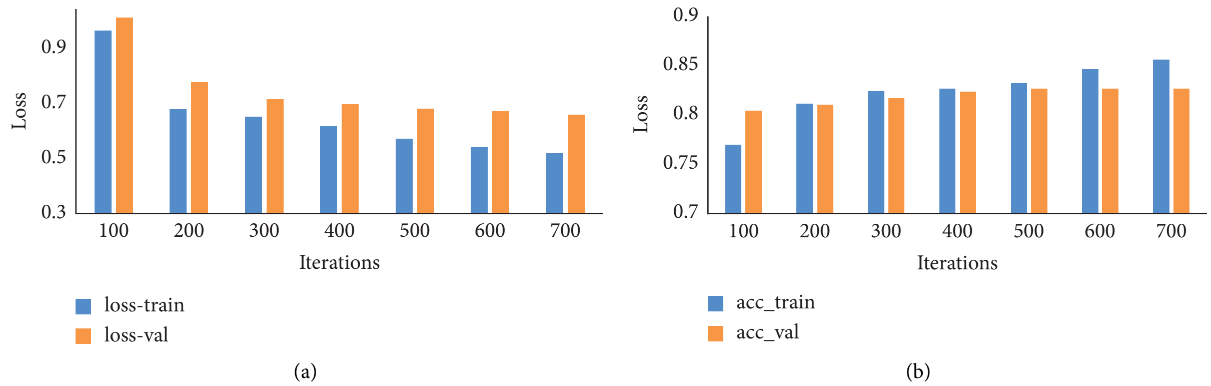

Figure 9: Experimental comparison of the effect of changing the iterations on loss function and accuracy. (a) Comparison of loss. (b) Comparison of acc.

Table 6: Ablation study of the key components of HN-DKG.

Model	MovieLens-1M for small amounts of data					
	Precision_train	AUC_train	Precision_eval	AUC_eval	Precision_test	AUC_test
HN-DKG	0.9233	0.9703	0.8934	0.9251	0.9218	0.9015
HN-DKG-drop1	0.923	0.913	0.882	0.9171	0.893	0.853
HN-DKG-drop2	0.91	0.897	0.873	0.918	0.875	0.87
HN-DKG-drop3	0.9101	0.9694	0.8435	0.9171	0.8423	0.9169

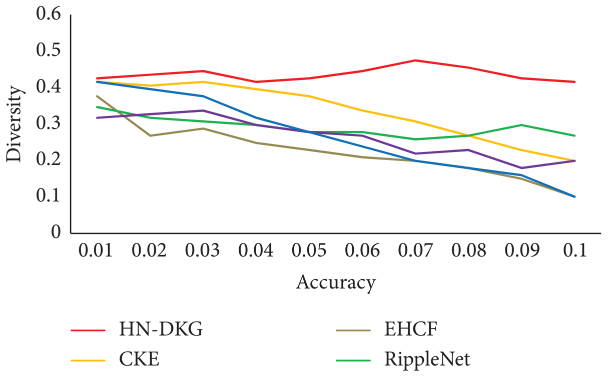

Figure 10: Relationship between accuracy and diversity of recommendation results under different models.

evaluation metrics of the training and verification model of HN-DKG-drop2 and HN-DKG-drop3 have also decreased. It is proved that the time warehouse mechanism and seed clustering method are useful for improving the performance of recommender system.

In this paper, the experiment proves the effectiveness of HN-DKG in improving the recommendation efficiency by comparing the parameters of two datasets. Ablation experiments were carried out to prove the effectiveness of virtual nodes, time warehouse mechanism, and node clustering factors. Therefore, it can be concluded that the proposed recommendation method based on multiplatform heterogeneous networks and dynamic knowledge maps considers the correct direction of factors and significantly improves the accuracy and diversity of results.

5. Conclusions and Future Work

Data sparsity, cold start, and data bias are the main factors affecting the performance of the recommender system. In order to improve the accuracy and diversity of the recommendations, we propose a recommendation approach based on heterogeneous networks and dynamic knowledge graphs. First, heterogeneous networks are built based on cross domain and multi-platform data to mine users' implicit preferences. After extracting knowledge maps from multimodal heterogeneous networks, a time warehouse mechanism is established. The GAT component is used to extract user preference features and calculate four types of attention weights. The additional relationship can increase the diversity of recommendation results and enhance the interaction relationship. The RippleNet algorithm is improved by using some mechanisms such as excellent seed clustering, random seed, and propagation blocking to improve the accuracy and increase the diversity of the recommendations, and further reduce the complexity of the algorithm.

The experimental results show that the proposed HN-DKG has the following characteristics:

(1) The basic knowledge graph is established through multi-modal heterogeneous networks to reduce the impact of selection bias, and at the same time, it is conducive to mining the potential preferences of users.

(2) The GAT component is used to calculate the attention weight of real-time multitasks, extract the short-term preference of users, enhance the feature relationship, and realize the timeliness of the recommendation approach.

(3) Excellent seed clusters, propagation blocking, and random seed mechanisms are used to reduce the impact of exposure deviation and improve the diversity of recommendations.

The HN-DKG proposed in this article can be applied in many scenarios which are related to recommendations, such as news feed, one-stop tourism and dining, public opinion guidance, and financial industry investment and social circle participation. In the future work, we will conduct research on more types of deviation problems and continue to study how to mine users' social behavior and potential preference characteristics according to the complex behavior of users in multiple platforms. At the same time, the combination of dynamic knowledge graph and deep learning algorithm is also the key to improve the recommender system.

Data Availability

The public data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors wish to acknowledge the support of the National Natural Science Foundation of China (Nos. 61902016 and 62271036), National 14th Five Year Plan Key R&D Projects (No. 2022YFB3305602-03), the National Key R&D Program of China (No. 2018YFC0807806-2), Beijing Natural Science Foundation General Project (No. 4232021), BUCEA Postgraduate Education and Teaching Quality Improvement Project (Nos. J2023002 and J2022005), and BUCEA Post Graduate Innovation Project (No. PG2023086).

References

[1] M. Zhang, G. Wang, L. Ren, J. Li, K. Deng, and B. Zhang, "METoNR: a meta explanation triplet oriented news recommendation model," *Knowledge-Based Systems*, vol. 238, Article ID 107922, 2022.

[2] B. Koo, H. Jeon, and U. Kang, "PGT: news recommendation coalescing personal and global temporal preferences," *Knowledge and Information Systems*, vol. 63, no. 12, pp. 3139–3158, 2021.

[3] H. Werneck, N. Silva, A. Pereira et al., "A reproducible POI recommendation framework: works mapping and benchmark evaluation," *Information Systems*, vol. 108, Article ID 102019, 2022.

[4] L. Zhang, Z. Sun, J. Zhang, Y. Wu, and Y. Xia, "Conversation-based adaptive relational translation method for next poi recommendation with uncertain check-ins," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7810–7823, 2023.

[5] B. Mulyawan, Vionelsy, and T. Sutrisno, "Product recommendation system on building materials shopping using FP-Growth algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 1007, no. 1, Article ID 12144, 2020.

[6] Y. Zhu, Q. Lin, H. Lu et al., "Recommending learning objects through attentive heterogeneous graph convolution and operation-aware neural network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 4178–4189, 2023.

[7] F. J. Tey, T. Y. Wu, C. L. Lin, and J. L. Chen, "Accuracy improvements for cold-start recommendation problem using indirect relations in social networks," *Journal of Big Data*, vol. 8, no. 1, pp. 98–118, 2021.

[8] N. Silva, D. Carvalho, A. C. M. Pereira, F. Mourão, and L. Rocha, "The pure cold-start problem: a deep study about how to conquer first-time users in recommendations domains," *Information Systems*, vol. 80, pp. 1–12, 2019.

[9] Y. Wang, "Research on recommendation algorithm of education platform for data sparsity," *Journal of Jiamusi University (Natural Science Edition)*, vol. 39, no. 6, pp. 137–139+166, 2021.

[10] B. Xu, "Research on interpretable recommendation method for sparse data," MA thesis, Sichuan University, Sichuan, China, 2021.

[11] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," in *Proceedings of the The world wide web conference*, San Francisco, CA, USA, May 2019.

[12] S. Zhu, J. Yang, J. Zhang, A. Bozzon, L. K. Huang, and C. Xu, "Recurrent knowledge graph embedding for effective recommendation," in *Proceedings of the 12th ACM conference on recommender systems*, pp. 297–305, Vancouver, Canada, October 2018.

[13] H. Wang, F. Zhang, J. Wang et al., "Ripplenet: propagating user preferences on the knowledge graph for recommender systems," in *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 417–426, Torino, Italy, October 2018.

[14] B. Yuan, Y. Liu, J. Y. Hsia, Z. Dong, and C. J. Lin, "Unbiased Ad click prediction for position-aware advertising systems," in *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 368–377, Brazil, September 2020.

[15] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," *Acm Sigir Forum*, vol. 51, no. 1, pp. 154–161, 2017.

[16] S. Wan, S. Yang, Y. Liu, J. Ding, D. Qiu, and C. Wei, "An image recommendation algorithm based on target alternating attention and user affiliation network," *Applied Sciences*, vol. 13, no. 7, p. 4389, 2023.

[17] Y. Lu, C. Shi, L. Hu, and Z. Liu, "Relation structure-aware heterogeneous information network embedding," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 4456–4463, 2019.

[18] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 357–370, 2019.

[19] J. Zhao, Z. Zhou, Z. Guan et al., "Intentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2347–2357, Anchorage, AK, USA, August 2019.

[20] J. Ding, Y. Quan, X. He, Y. Li, and D. Jin, "Reinforced negative sampling for recommendation with exposure data," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*, pp. 2230–2236, Macao, China, August 2019.

[21] X. Wang, Y. Xu, X. He, Y. Cao, M. Wang, and T. S. Chua, “Reinforced negative sampling over knowledge graph for recommendation,” in *Proceedings of the Web Conference 2020 (WWW ’20)*, pp. 99–109, Taipei, Taiwan, April 2020.

[22] X. Wang, R. Zhang, Y. Sun, and J. Qi, “Doubly robust joint learning for recommendation on data missing not at random,” *International Conference on Machine Learning*, vol. 97, pp. 6638–6647, 2019.

[23] W. Zhang, W. Bao, X. Y. Liu et al., “Large-scale causal approaches to debiasing post-click conversion rate estimation with multi-task learning,” in *Proceedings of the Web Conference 2020*, Taipei, Taiwan, April 2020.

[24] L. Yang, Z. Xiao, W. Jiang, Y. Wei, Y. Hu, and H. Wang, “Dynamic heterogeneous graph embedding using hierarchical attentions,” in *Proceedings of the Advances in Information Retrieval: 42nd European Conference on IR Research*, pp. 425–432, ECIR, Lisbon, Portugal, April 2020.

[25] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” *Advances in Neural Information Processing Systems*, vol. 1, pp. 577–585, 2015.

[26] Y. Gong and Q. Zhang, “Hashtag recommendation using attention-based convolutional neural network,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 577–585, IJCAI’16, New York, NY, USA, July 2016.

[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *Proceedings of the ICLR 2018*, Vancouver, Canada, May 2018.

[28] X. Li, X. Zhang, P. Wang, and Z. Cao, “Web services recommendation based on Metapath-guided graph attention network,” *The Journal of Supercomputing*, vol. 78, no. 10, pp. 12621–12647, 2022.

[29] Z. Zhang, J. Huang, and Q. Tan, “Association rules enhanced knowledge graph attention network,” *Knowledge-Based Systems*, vol. 239, Article ID 108038, 2022.

[30] W. Zhang, W. Bao, X. Y. Liu et al., “Large-scale causal approaches to debiasing post-click conversion rate estimation with multi-task learning,” in *Proceedings of the Web Conference 2020*, pp. 2775–2781, Taipei, Taiwan, April 2020.

[31] S. Guo, L. Zou, Y. Liu et al., “Enhanced doubly robust learning for debiasing post-click conversion rate estimation,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 275–284, Canada, July 2021.

[32] Z. Huang, Y. Cheng, R. Cheng, Y. Sun, N. Mamoulis, and X. Li, “Meta structure: computing relevance in large heterogeneous information networks,” in *Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining*, pp. 1595–1604, San Francisco, CA, USA, August 2016.

[33] X. Yu, X. Ren, Q. Gu, Y. Sun, and J. Han, “Collaborative filtering with entity similarity regularization in heterogeneous information networks,” in *Proceedings of the International Joint Conference on Artificial Intelligence Heterogeneous Information Network Analysis 27*, Beijing, China, August 2013.

[34] C. Shi, X. Kong, Y. Huang, P. S Yu, and B. Wu, “Hetesim: a general framework for relevance measure in heterogeneous networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2479–2492, 2014.

[35] H. Wang, F. Zhang, X. Xie, and M. Guo, “DKN: deep knowledge-aware network for news recommendation,” in *Proceedings of the 2018 world wide web conference*, Lyon, France, April 2018.

[36] C. Shi and P. S. Yu, *Heterogeneous Information Network Analysis and Applications*, Springer International Publishing, Cham, Switzerland, 2017.

[37] Y. Wang, Y. Wu, and J. Wu, “Research on recommendation model for multi label system based on heterogeneous network,” *Journal of Software*, vol. 28, no. 10, pp. 2611–2624, 2017.

[38] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, “Leveraging metapath based context for top-n recommendation with a neural co-attention model,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1531–1540, London, UK, August 2018.

[39] W. Feng and J. Wang, “Incorporating heterogeneous information for personalized tag recommendation in social tagging systems,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD ’12)*, pp. 1276–1284, Beijing, China, August 2012.

[40] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, “Semantic path based personalized recommendation on weighted heterogeneous information networks,” in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 453–462, Melbourne, Australia, October 2015.

[41] X. Yu, X. Ren, Y. Sun et al., “Recommendation in heterogeneous information networks with implicit user feedback,” in *Proceedings of the 7th ACM conference on Recommender systems*, pp. 347–350, Hong Kong, China, October 2013.

[42] 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*, Anchorage, AK, USA, August 2019.

[43] X. Wang, “Research on relational metric social recommendation algorithm based on graph attention network,” MA thesis, Beijing Jiaotong University, Beijing, China, 2021.

[44] Y. Zeng and S. Liu, “Research on recommendation algorithm of Graph attention Network based on Knowledge graph,” *Journal of Physics: Conference Series*, vol. 2113, no. 1, Article ID 12085, 2021.

[45] S. Wan and Y. Liu, “A security detection approach based on autonomy-oriented user sensor in social recommendation network,” *International Journal of Distributed Sensor Networks*, vol. 18, no. 3, Article ID 155013292210824, 2022.

[46] Y. Luo, “Research on file recommendation methods based on Cn RippleNet,” MA thesis, Northwest University for Nationalities, Lanzhou, China, 2022.

[47] C. Shi, “Research on improved RippleNet recommendation method,” MA thesis, Huazhong University of Science and Technology, Wuhan, China, 2022.

[48] Y. Luo, B. Sha, and T. Xu, “A recommended method based on the weighted RippleNet network mode,” *Journal of Physics: Conference Series*, vol. 2025, no. 1, Article ID 12011, 2021.

[49] Y. Wang, L. Dong, Y. Li, and H. Zhang, “Multitask feature learning approach for knowledge graph enhanced recommendations with RippleNet,” *PLoS One*, vol. 16, no. 5, Article ID e0251162, 2021.

[50] X. S. Ji, Y. B. Liu, and L. M. Luo, “Similarity measurement based on user interest in collaborative filtering,” *Journal of Computer Applications*, vol. 30, no. 10, pp. 2618–2620, 2010.

[51] J. Yang and L. Gao, “Application of prospective selection sampling algorithm in product recommendation,” *Computer Applications*, vol. 9, no. 9, pp. 2177-2178, 2005.

[52] J. Xu, X. Wang, H. Zhang, and P. Lv, "Heterogeneous and clustering-enhanced personalized preference transfer for cross-domain recommendation," *Information Fusion*, vol. 99, pp. 101892–102535, 2023.

[53] T. Man, H. Shen, X. Jin, and X. Cheng, "Cross-domain recommendation: an embedding and mapping approach," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, pp. 2464–2470, Melbourne, Australia, August 2017.

[54] C. Chen, M. Zhang, Y. Zhang, W. Ma, Y. Liu, and S. Ma, "Efficient heterogeneous collaborative filtering without negative sampling for recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 19–26, 2020.

[55] F. Zhang, J. N. Yuan, D. Lian, X. Xie, and W. Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 353–362, San Francisco, CA, USA, August 2016.

[56] X. He and T. S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 355–364, Tokyo, Japan, August 2017.

[57] B. Antoine, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)*, pp. 2787–2795, Lake Tahoe, NV, USA, December 2013.

[58] D. Bolya, C. Y. Fu, X. Dai, P. Zhang, and J. Hoffman, "Hydra attention: efficient attention with many heads," in *Proceedings of the Computer Vision– ECCV 2022 Workshops*, ECCV, Tel Aviv, Israel, October 2022.

[59] M. Pan, "Research on social recommender system based on graph attention network," MA thesis, Northeast Normal University, Changchun, China, 2021.

[60] N. Hurley and M. Zhang, "Novelty and diversity in top-n recommendation--analysis and evaluation," *ACM Transactions on Internet Technology*, vol. 10, no. 4, pp. 1–30, 2011.

[61] S. Vargas and P. Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in *Proceedings of the fifth ACM conference on Recommender systems (RecSys '11)*, pp. 109–116, Chicago, IL, USA, October 2011.