A Dual-End Recommendation Algorithm Integrating User Intent and Knowledge-Aware Attention Networks

Zijie Ji
School of Electronic and Information Engineering, Anhui Jianzhu University
China
j15255553869@163.com

Teng Lv
School of Big Data and Artificial Intelligence, Anhui Xinhua University
China
LT0410@163.com

Yi Yu
School of Electronic and Information Engineering, Anhui Jianzhu University
China
yy1548406485@163.com

Abstract: *The existing knowledge graph-based recommendation models often lack a fine-grained consideration of collaborative information between users and items and overlook the high-order semantics and structural relationships within the graph paths. To address these issues, a dual-end recommendation algorithm integrating User Intent and Knowledge-Aware Graph Attention Networks (UIKGAN) is proposed. On the user end, the intent behind user-item interactions to refine the representation of collaborative information is modeled. By propagating relationship paths, UIKGAN aggregates deeper semantic and structural information from the knowledge graph to more accurately capture the extended representation of user intent and behavior patterns. On the item end, UIKGAN embeds and aggregates high-order neighboring triplet information using a knowledge-aware attention mechanism, enriching the feature representation of items. Additionally, this paper introduces an independence modeling module to optimize the loss function, providing better interpretability of user intent. Experiments were conducted on three public datasets, including comparative experiments with seven baseline models, ablation studies, hyperparameter sensitivity experiments, and sparse data issue analysis. The experimental results demonstrate that the UIKGAN model outperforms other baselines in overall performance, improving recommendation accuracy while effectively alleviating the issue of dataset sparsity.*

Keywords: *Recommendation model, user intent, knowledge-aware attention, dual-end recommendation.*

Received August 28, 2024; accepted March 3, 2025
https://doi.org/10.34028/iajit/22/3/7

1. Introduction

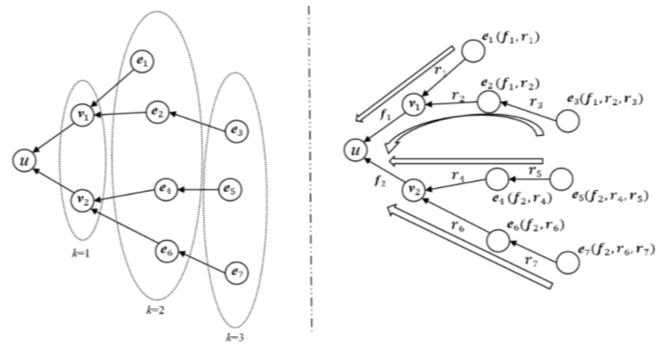
In the era of big data, recommendation systems have become an indispensable tool in people's daily life [29]. Recommender systems use the interaction information between users and items to capture users' historical interests, calculate the interaction possibilities between users and different items, and then recommend their preferred content for users to meet personalized needs [4]. The data used in recommendation systems includes the user's own attribute information, the user's historical operating behaviour, candidate item information, and contextual scenarios, etc., which can be divided into structured data, semi-structured data and unstructured data [6]. In recent years, recommendation algorithms combining knowledge graphs have been widely used by the industry, and have achieved remarkable results in improving the problems of cold start and data sparsity in traditional recommendation systems. However, due to the heterogeneous information network attributes in the knowledge graph, many models still have certain limitations in practical scenarios [28].

Current recommendation models based on embedding methods enrich item attribute descriptions by constructing auxiliary information graphs for various types of items. Knowledge embedding techniques are used to represent entities and relationships in the graph as low-dimensional vectors, preserving the original structural information of the graph. However, the drawback of these methods is that they overlook the semantic relationships between entities and do not thoroughly consider the interaction information between users and items [26].

Historical user-item interaction data, also known as collaboration information, contains complex implicit feedback. For example, in the movie recommendation context, the target user chose to watch the movie Kung Fu, stemming from his preference for the movie's star, Stephen Chow; another user also watched the movie, but chose it because the movie genre is action. Although the two users' behaviours were the same, the reasons for their choices were different, which suggests that the users made the same choices based on different intentions. Different intents reflect different behavioural patterns of users, and by analyzing such intent tendencies, the implicit interaction information between users and items can be mined at a finer granularity [27], so as to better personalize and accurately recommend for users from their perspective.

Most recommendation models based on the path approach utilize the structural characteristics of network links in the knowledge graph to construct paths for the relationships between nodes, in order to mine the deep underlying semantics between entities, and at the same time differentiate and analyze different types of association paths [2]. Recommendation models based on the path approach have excellent interpretability and can effectively address the shortcomings of previous graph neural network-based recommendation models [22]. This type of model is usually an end-to-end model, and the aggregation of important nodes is achieved through the iterative accumulation of network hierarchies [17]. The flow of processing graph data is shown in Figure 1-a), which aggregates a total of three jumps of neighbour node information for prediction, but ignores the relationship information between entities, and only uses the relationship edges as the index of path connections.

Users' potential preferences are often embedded in the relational information on the path, as shown in Figure 1-b). Therefore, effectively combining the path information and the relationship information on the edges between nodes in the knowledge graph with the user's collaborative information in the recommendation model plays an important role in improving the accuracy of a recommendation system.



a) Domain node aggregation based on neutral network recommendation models.

b) Domain node aggregation based on path policy recommendation models.

Figure 1. Comparison of neighbourhood node aggregation processes for two types of recommendation models.

In order to better address the shortcomings of the above models, this paper proposes a User Intent and Knowledge-Aware Graph Attention Network (UIKGAN) model. This model utilizes a propagation-based recommendation approach that incorporates the advantages of the graph embedding strategy to enhance the feature representation of graph entities and relationships, and at the same time solves the shortcomings of this strategy that ignores the higher-order connectivity among entities. On the user end, the advantages of the path-based approach are combined to model the user's intention tendency and analyze the user's collaboration information in more detail. Meanwhile, during the link propagation process, the relationship-dependent information on the path is combined with the collaboration information to effectively explore the users' potential intent preferences and behavioural patterns. On the item end, graph attention mechanism is used to aggregate the higher-order nearest-neighbour information of the items. UIKGAN has good interpretability and demonstrates strong recommendation performance on sparse datasets by utilizing rich user and item representations to assist predictions.

The organization of this paper is as follows. In section 2, existing technical knowledge such as deep learning, graph neural networks, knowledge graphs, and graph attention mechanisms were described. In section 3, the structure of the model was introduced, and the role of each module of the model was analyzed one by one. In the section 4, the results and performance evaluation are described of our model UIKGAN. Finally, section 5 summarized the paper and proposed future research directions.

2. Related Work

Current mainstream recommendation algorithms are mainly content-based recommendation, collaborative filtering-based recommendation [1] and deep learning-based hybrid recommendation. Content-based methods and collaborative filtering-based methods are often limited to specific recommendation scenarios. For example, the content-based recommendation model needs enough historical user behaviour information for modelling, and the collaborative filtering-based recommendation model also needs a huge amount of rating data in order to train a good recommendation effect. Therefore, the dataset to be processed by the model needs to contain rich attribute information and interaction data. In addition, modeling of these two types of methods is overly simplistic, making it difficult to accurately uncover users' potential needs from large datasets. To address the shortcomings of these methods, deep learning techniques can be employed to integrate the two approaches through a fusion strategy, resulting in a hybrid recommendation method based on deep learning, which combines their strengths. At the same time, the algorithm construction and recommendation effect can be improved as a whole. This is also an important theoretical basis for the dual-end model in this paper.

2.1. Deep Learning Based Hybrid Technologies

Deep learning-based recommendation methods introduce knowledge graph information into the user-item relationship in order to effectively alleviate the cold-start problem, and at the same time endow the model with better interpretability. For example, National Infocomm Competency Framework (NICF) [30] is a deep learning-based neural interactive filtering recommendation algorithm. To capture user interests, the model NICF employs multi-channel stacked neural networks to represent exploration strategies. It learns directly from feedback data and continuously updates the user interest model based on new data to enhance the performance of interactive recommendations. Lin *et al.* [9] proposed a deep learning-based social recommendation model GNN-DSR with dynamic and static representations. GNN-DSR models the immediate changes of user interests while considering the static features of long-term interactions, and uses multi-dimensional inference of potential features to make reasonable predictions, which is applicable to realistic social scenarios.

2.2. Graph Neural Network Related Techniques

Graph Convolutional Networks (GCN) [8] is applied to graph structure to allow nodes in the graph to be better associated with their neighbouring nodes. GCN's characteristic is recursive accumulation, allowing target nodes to continuously acquire important neighbor information even from distant paths. The drawback of GCN lies in its scalability, where with the accumulation of iterative layers, the number of neighboring nodes also exponentially increases. This heavily challenges the computational capacity of the model, hence requiring careful control over the depth of iteration when applying the algorithm.

To perform reasonable sampling on graph structured data, GraphSage [5] proposed an efficient algorithm based on nearest neighbor sampling. By sampling, only a subset of its neighbors for message passing on each node, GraphSage not only significantly reduces the storage and computational complexity of graph structured data during computation, but also avoids the memory bottleneck problem of traditional full graph training methods.

RippleNet [18] is a ripple network model used to propagate user preferences in graphs, and to propagate and mine user preference information in knowledge graphs. This model starts from the user's initial points of interest, propagates interest signals layer by layer in the graph, and constructs a recursive propagation network of user preferences along the path of the graph. The propagation mechanism of RippleNet can not only capture direct associations between users and items, but also explore deeper indirect relationships and potential semantic associations through multi-level transmission processes.

Knowledge Graph Convolutional Networks (KGCN) [20] utilized the message passing mechanism of GCN and combined it with classical recommendation models to design an efficient recommendation method. Through GCN, KGCN can aggregate information from neighboring nodes to achieve deeper modeling of user interests and project features, thus performing well in recommendation tasks.

Knowledge-aware Graph Neural Networks with Label Smoothness regularization (KGNN-LS) [19] has made improvements to KGCN, particularly in addressing cold start issues. The KGNN-LS model introduces a label smoothing regularization mechanism, which constrains the edge weight matrix of the graph to ensure the smoothness and consistency of label information propagation in the graph, thereby more effectively expanding entity labels and feature information.

2.3. Knowledge Graph Based Recommendation Algorithms

Knowledge graphs, as a kind of auxiliary information, possess rich information resources and excellent semantic structure representation. In the graph, the information consists of multi-relational graphs composed of various types of entities (nodes) and different edges (relations), which implies a rich ternary structure, and can be efficiently displayed for structured data. To improve the knowledge graph operation efficiency. researchers use Knowledge Graph Embedding (KGE), which describes the specifics of entities and relationships in the graph in terms of vectors, and obtains their respective vector representations through computational training between ternaries.

Traditional recommendation models are often limited to the correlation information between user-items, ignoring the intrinsic correlation between the user's own information and the item's own attributes. A dataset incorporating knowledge graphs can introduce rich entity semantic extensions in modelling user-item interactions. Knowledge Graph Attention neTwork model (KGAT) [21] combines user-item interactions and the corresponding graph information into a Collaborative Knowledge Graph (CKG), and through the attention mechanism recursively dissemination of information. However, the defects of this model are that if a new interacting user enters, it is necessary to rebuild the CKG and retrain the whole model, which results in computational redundancy; and CKG module treats the item entities from user historical interactions and the related entities in the graph as isomorphic nodes, but in reality, they are in different latent spaces, which can easily cause ambiguity. KGAT-based Collaborative Knowledge Awareness model (CKAN) [23] constructed a heterogeneous propagation module, entities in the user's historical interactions and entities in the graph are regarded as information in different spaces, and share different weight information through natural combination. In addition, CKAN introduces an attention mechanism and a multi-layer co-propagation module to effectively fuse the collaborative signals and knowledge associations in order to improve the performance of the model recommendation. Knowledge-Aware User Preference Model (AKUPM) [13] takes into account the differences in the characteristics of the entities in different relationships, and projects the entities to the space of the relationships that they are connected to, in order to reflect the independence of the characteristics of the different relationships. Hierarchical Knowledge Graph embedding model for personalized recommendation (HAKG) [12] encodes subgraphs through hierarchical attention and thus generates effective subgraph embeddings to enhance the prediction of user preferences. A model Knowledge enhanced Graph Neural Network (KeGNN) [10] for interpretable recommendations combines neural networks and symbolic reasoning, while introducing a knowledge enhancement layer in KeGNN to adjust the model's predictions by learning clause weights, which in turn improves the accuracy of predictions.

2.4. Graph Attention Mechanisms

The traditional GCN model has many limiting problems, such as the inability to deal with dynamic graph problems as well as the fact that GCN can only deal with homogeneous graphs and cannot deal with multimodal or heterogeneous graphs. In order to solve such problems, Graph Attention Network (GAT) proposes a weighted summation of neighbouring node features with an attention mechanism, where the weights of the neighbouring node features are completely dependent on the node features and independent of the graph structure.

The core difference between GAT and GCN is how to collect and sum the feature representations of neighbour nodes with distance 1. The graph attention model GAT replaces the fixed normalization operation in GCN with an attention mechanism. Essentially, GAT simply replaces the original GCN normalization function with a neighbour node feature aggregation function using attention weights.

In GAT, each node in the graph can be assigned different weights based on the features of its neighbouring nodes. With the introduction of attention mechanism, it is only relevant to the neighbouring nodes, i.e., nodes sharing edges, without the need to get information about the whole graph.

GAT improves GCN by introducing an attention mechanism that allows nodes to assign weights based on the importance of their neighbouring nodes. However, GAT also has its limitations. First, GAT has a high computational complexity because it needs to compute the attention scores between each pair of nodes. In addition, GAT likewise does not handle dynamic and heterogeneous graphs well. Moreover, although GAT enhances the expressive power of the model through the attention mechanism, this also increases the complexity and instability of the model.

In summary, the goal of this paper is to propose a dual-ended recommendation algorithm that integrates UIKGAN, in response to the problem that existing knowledge graph-based recommendation models lack careful consideration of collaborative information about the history of user-item interactions, and ignore the higher-order semantic and structural relationships in the paths of the graphs. On the user end, user intent is modelled by a propagation-based approach and propagated along the graph path, effectively combining the relationship-dependent information on the path with the collaborative information, in order to better mine the user's potential intent preferences and behavioural patterns. On the item end, knowledge-aware graph attention is applied to enhance the feature representation of entities and relationships, which solves the problem of traditional recommendation models ignoring the higher-order connectivity between entities and enriches the attribute features of items. Finally, the vectors of both user-end and item-end are fed into the prediction stage, and the independence modelling loss is introduced in the loss function module to obtain good prediction recommendation results.

3. UIKGAN Model

This model adopts a dual-end structure, addressing recommendation tasks from both the user and item perspectives to capture finer-grained collaborative information and rich high-order semantics. The user side models user intentions and extracts preference features underlying interactions, while the item side leverages a knowledge-aware attention mechanism to aggregate high-order neighbor information, thereby enhancing item representation.

3.1. Notation and Definition of UIKGAN Model

The data used in the model UIKGAN is structured data, including the collaboration information of the user and the item and the corresponding knowledge graph data of the item entity, as well as the knowledge graph of the collaboration information.

1. Information on user-item collaboration.

In the specific recommendation scenario, define the relevant notation as follows: Let the set of users as $U=\{u_1, u_2, ..., u_m\}$, the collection of items is set to $V=\{v_1, v_2, ..., v_n\}$, User-Item Interaction data, also known as collaboration information, is represented as User-Item bipartite graph $G_1=\{(u, y_{uv}, v)\ u \in U, v \in V\}$, where $y_{uv}=1$ indicates that there is an interactive operation between user u and item v, such as clicking, browsing, etc.; otherwise $y_{uv}=0$.G_1 is referred to as collaboration information later in this paper.

2. Knowledge graph data.

A knowledge graph corresponding to collaborative information in a dataset contains attribute information associated with an item, such as information about the attributes of the item or information about related general knowledge. Define the graph data as $G_2=\{(h, r, t)\ h, t \in E, r \in R\}$, where E and R are the sets of entities and relationships in the knowledge graph, respectively. G_2 consists of the set of entity-relationship-entity triples

$S(h, r, t)$, where $h, t \in E$, and $r \in R$. G_2 is referred to as the original knowledge graph in this paper.

3. Collaborative information knowledge graph.

Alignment operation is performed between the set V of items in the user-item bipartite graph and the set E of entities in the knowledge graph to realize the natural connection between the user collaboration information and the original graph information, and the association relationship between them is defined as the set $D=\{(v,e)|v \in V, e \in E)\}$, and define collaborative information knowledge mapping as $G=\{(h, r, t)|h, t \in E', r \in R'\}$, where $E'=E \cup V$, $R'=R \cup y_{uv}$. G is referred to as interaction knowledge graph in this paper.

The model algorithmically predicts the probability $\hat{y}_{uv}$ of a user u interacting with an item v that did not occur.

Table 1 is a symbolic introduction of the model UIKGAN.

Table 1. Notations of UIKGAN model.

Notation	Meaning
U	set of users
u	user
V	collection of items
v	item
G_1	User-Item bipartite graph (collaboration information)
y_{uv}	Indicator of interactive operation
G_2	graph data
E	set of entities
R	sets of relationships
S(h,r,t)	set of entity-relationship-entity triples
D	association relationship
G	collaborative information knowledge mapping (interaction knowledge graph)

3.2. Model Frame of UIKGAN

Figure 2 shows the overall framework diagram of our proposed model UIKGAN. Data processing starts from both ends, as follows: handling collaborative knowledge graph data and conducting deep exploration through paths on the user end; processing original graph data and aggregating essential item attribute information on the item end.

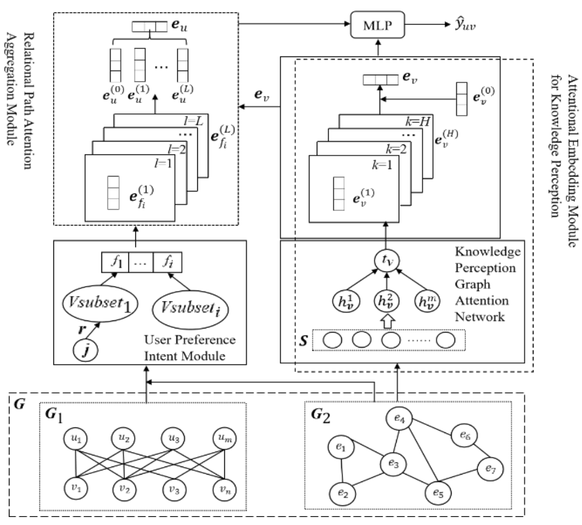


Figure 2. The overall framework of UIKGAN model.

The overall framework of the UIKGAN model is as follows: on the user end, it contains the user intention tendency module and the relational path attention aggregation module; on the item end, it contains the knowledge-aware attention embedding module; and the final Multi-Layer Perception (MLP) prediction module.

The main contribution of this paper is as follows:

1. On the user end, the items that users have interacted with historically are divided into I sets, and the user intent tendency module is constructed to describe user collaboration information at a fine-grained level, and differentiate the potential preference information of users on different propagation paths through different intents; By leveraging attention aggregation on relational paths, deeper layers of user intent expansion can be mined in the interaction graph.
2. On the item end, use the knowledge-aware graph attention network to aggregate higher-order item neighbourhood attribute triplet entities, thus enriching the aggregation vector of the item; at the same time, pass the item vector back to the user end, and aggregate it with the intention tendency factor vector through the attention mechanism to obtain the final user aggregation vector.
3. Independence modelling is introduced into the loss function to emphasize the independence between different user intentions and give the model better interpretability. Finally, the aggregated vectors of users and items obtained from both ends are used for recommendation prediction by MLP.

3.3. User Intention Tendency Module

When dealing with user-item collaboration information, classical recommendation methods tend to simply regard the user's interaction behaviour as a relational connection with the item, without digging deeper into the hidden motives behind the interaction behaviour. These motives can be understood as the differences caused by the user's focus on the items, i.e., the different intentions based on their own preferences (e.g., in the movie recommendation scenario, the same user will prefer a certain movie star to watch a different movie starred by the actor, and similarly, will choose to watch another movie filmed by a certain director because of his preference). From this, it can be inferred that among users with similar preference tendencies, the probability of their underlying interests is also similar. Based on this idea, combined with the idea of path-based recommendation methods, the user intent preference factor is introduced to simulate the user's preference intention to make choices in real scenarios, thus refining the motivation behind user preferences to enrich user-end modelling.

Figure 3 shows the modelling of the user's intentional tendency. The set of items that have interacted with the target user is evenly divided into subsets of I, referred to as interaction subsets ($Vsubset_i$),

i=1, 2, …, I. The edges of these subsets of links to the user contain information about the user's historical interaction behaviour, and each subset of linking relationships corresponds to an intentional tendency, then each of the subsets corresponds to intention tendency factors, and these factors are defined as $f_1, f_2, ..., f_i$, i=1, 2, …, I. Assign intent attributes to the interaction relationships between users and items, where the intention tendency factor can be described as the degree of attention given to different interaction relationships.

User-item collaboration information $(u, v) \in G_1$ can be extended as user Intention Orientation Graph (G_{IO}), where $G_{IO}=\{(u, f_i, v) | i=1, 2, ..., I\}$, $G_{IO} \in G$, which can be considered as the set of behavioural intentions that user u generates for the item entities in the interaction graph under the action of tendency f.

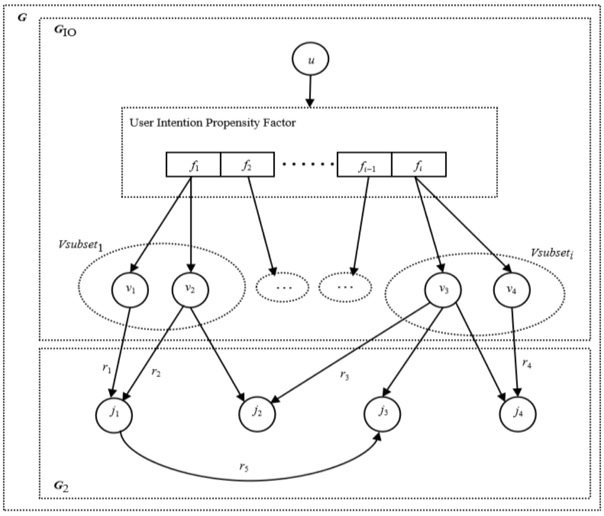


Figure 3. User intention tendency module.

In previous graph-based embedding strategies, intent information can be expressed through potential vectors, but it is difficult to explicitly identify the semantics of each intent. A simple solution is to associate each intention with a relation in the knowledge graph, such as the embedding model KTUP [3]. However, this model only considers a single relationship pattern and ignores the complex interactions and combinations in the relationship, thus failing to refine the implicit information of user intentions. This paper attempts to propagate user interest features through paths in interaction graph, considering the close connection between items in the user's collaborative information and the entities and relationships in the original graph. As path $u \to v_1 \to j_1 \to j_2$ shown in Figure 3, u reaches v_1 through f_1, entities j_1 and j_2 can be reached sequentially along relations r_1 and r_5. It is evident that the user's intent inherently determines the direction of the path.

Therefore, mining the potential interest entities and inter-entity relationships on each path in the interaction graph can accurately model the user's intention preference factor. Define the first-order vector representation of the factor as:

$$e_{fi}^{(1)} = \frac{1}{|N\_v|} \sum_{(j,r) \in N_v} e_j \odot e_r, i = 1,2,\dots,I \quad (1)$$

where e_j and e_r correspond to the vector representati of entities and relations in the interaction graph, and interact with each other through dot product operations. By using the triplet $S_v(v, r, j)$, different user intent tendency factors influence the retrieval of item entities and the associated paths between entities. For example, the triplet corresponding to the factor f_1 in Figure 3 is (v_1, r_1, j_1), which indicates that the user, under the influence of the tendency f_1, selects the item v_1 while at the same time generates an intention for j_1 in the interaction graph through the relation r_1.

In Equation (1), N_v denotes the set of triples of all first-order relational paths ending in v in the propagation path of the interaction graph, defined as:

$$N_v = \{(j, r) | (v, r, j) \in G, v \in Vsubset_i, i = 1,2,\dots,I\} \quad (2)$$

The potential preference of users for items is reflected in the modelling of the intention preference factor. If similar recommendation results come from different combinations of relationships, it suggests that the target user's intention can be inferred from another user's tendency. However, if an intent can be inferred from other intents, it may be redundant and no longer has the ability to characterize user behaviour.

Therefore, intent independence modelling is introduced, which uses the distance correlation coefficient as a regularizer to minimize the distance correlation of a user's tendency by measuring the linear relationship between any two intent vectors, thus reducing the dependency between different intents. This independence modelling can effectively reduce the redundancy of data and improve the training efficiency of the model with the following equation:

$$L_{TEN} = \sum_{f, f' \in G_{IO}, f \neq f'} DIS(e_f, e_{f'}) \quad (3)$$

where $DIS(\cdot)$ is the correlation between the distances between the two intentional tendency factors e_f and $e_{f'}$.

$$DIS(e_f, e_{f'}) = \frac{dCov(e_f, e_{f'})}{\sqrt{dVar(e_f) \cdot dVar(e_{f'})}} \quad (4)$$

where $dCov(\cdot)$ denotes the distance covariance between the two intentional representations, and $dVar(\cdot)$ is used to measure the distance variance of each intentional representation. By integrating this module into the loss function, the models in this paper can be optimized. While encouraging divergence on different intentions, giving intentions clear boundaries grants better interpretability on the user end for modelling user intentions.

3.4. Relational Path Attention Aggregation Module

The classical collaborative filtering-based approach provides a better description of user preferences. If multiple users have interaction behaviours for the same item, these users constitute user collaborative neighbours; if multiple items are interacted by the same user, these items constitute collaborative neighbours of the items. The potential preference of users or the attribute characteristics of items can be directly reflected by the information of interacting collaborative neighbours, i.e., the history of user-item interactions can reflect the preference of users, and users who make similar behaviours have similar tendency to choose items.

The modelling of relational paths in the interaction graph refers to this setting by considering a user's historical interest as a pre-existing characteristic of the user, using historical interaction information (intentional collaboration information) that combines the user's intentional tendency factor to capture the user feature representation at a finer granularity level, and aggregating intentional correlation information on the paths through the attentional mechanism, to construct a vector of first-order representations of the user's u as:

$$e_u^{(1)} = g_{IO}(\{(e_u^{(0)}), e_{fi}^{(1)}), e_v\ )|(f_i, v) \in N_u, v \in Vsbset_i, i = 1,2,\ldots,I\}) \tag{5}$$

where the aggregation function $g_{IO}(\cdot)$ is the node neighbourhood information under the aggregation intention tendency, and $N_u=\{(f_i, v)|(u, f_i, v)\in G_{IO}, i=1, 2, ..., I\}$ is the set of collaboration information with the target user u intention. The aggregation function $g_{IO}(\cdot)$ is formally expanded as by the following equation:

$$e_u^{(1)} = \frac{1}{|N_u|} \sum\nolimits_{(f_i,v)\in N_u)} \beta(u,f) e_{f_i}^{(1)} \odot e_v \tag{6}$$

where e_v is the final aggregated vector at the item end, which can be used as the influence factor of the user vector; e_{f_i} and e_v are computed through the dot product operation, which emphasizes the user's intention to interact with the potentially preferred item under the tendency f_i; through the aggregation operation, the semantic information of the user under different intention tendency factors and the information of the item features of the user's historical interactions are integrated, which in turn enriches the user's preference vector representation; $\beta(u, f)$ is the attention score, different intention tendencies influence different behavioural patterns of the user, so this attention mechanism $\beta(u,f)$ is introduced to differentiate the influence of the intention, this attention score is defined as follows:

$$\beta(u,f) = \frac{exp(e_{f'}^T e_u^{(0)})}{\sum_{(f,f'\in G_{IO}} exp(e_{f'}^T\ e_u^{(0)})} \tag{7}$$

where $e_u^{(0)}$ is the user's ID embedding, the higher the attention score, the stronger the degree of influence of the factor on the user.

The path can originate from items preferred by the user and extend to entities where interest diffusion is observed.

Propagation along the relational paths can better aggregate important nearest-neighbour information in the interaction graph and capture the higher-order semantic information of entity-relationship interactions. If the depth of the relational path of the item-entity link in the interaction graph is L, the related directed path can be embodied as $j^L \xrightarrow{r^L} \cdots \rightarrow j^2 \xrightarrow{r^2} j^1 \xrightarrow{r^1} v\ (v \in Vsubset_i)$. Based on Figure 3, it can be seen that the path connection between the item and the entity in the interaction graph is bi-directional, and the path can be initiated by the user's preferred item to reach the endpoint of the entity where the interest is spreading; similarly, it can be propagated in reverse direction, starting from the entity endpoints. The same can be done from the endpoint of the entity for backward propagation. According to the definition of Equation (1), the formula of L-order user intention tendency factor is calculated as follows:

$$e_{f_i}^{(L)} = \frac{1}{|N_i^L|} \sum e_{j^1} \odot e_{r^1} + e_{j^2} \odot e_{r^2} + \cdots + e_{j^L} \odot e_{r^L} \tag{8}$$

where N_i^L is the set of L-order directed propagation paths. By performing dot product operations between entities and relations in the interactive knowledge graph and conducting cumulative aggregation, implicit relationships between entities are effectively mined. This approach preserves the deep semantic information along the propagation paths, facilitating the modeling of higher-order user intent preference factors.

After passing the generation result of Equation (8) into the operation of Equation (6), the L-order vector representation of user u is obtained by aggregating the higher-order item vectors generated at the item end as:

$$e_u^{(L)} = g_{IO}\left(\left\{\left(e_u^{(L-1)}, e_{f_i}^{(L)}, e_v\right) | (f_i, v) \in N_u, v \in Vsubset_i, i = 1,2,\ldots,I\right\}\right) \tag{9}$$

Thus, the final aggregated vector of user u is obtained, embedding the intentional tendency information in the path with the higher-order semantic interaction information in the interaction graph:

$$e_u = e_u^{(0)} + \cdots + e_u^{(L)} \tag{10}$$

3.5. Attention Embedding Module Based on Knowledge Perception

On the item end, the entity information in the original knowledge graph that aligns with the user's historical interest items is analyzed. These entities and their relations provide supplementary information for the user's historical interests. By integrating the idea of propagation-based recommendation methods [25], user preferences are propagated along the link paths in the graph. During the traversal process, graph attention is utilized to aggregate important neighborhood information at different levels. Applying this modelling to the item side enriches the item attribute descriptions, which in turn generates vector representations of items at a finer granularity. The following shows the specific item-side aggregation process:

1. Definition of propagation in the original graph.

The gradual propagation of knowledge associations along the links in the original graph allows for the acquisition of extended entities and their corresponding sets of triples that are more distant from the initial item entity. The set of extended entities is defined as follows:

$$\mathcal{E}^k = \{t|(h,r,t) \in G_2, h \in \mathcal{E}^{k-1}\}, k = 1,2,\cdots,H \quad (11)$$

where the parameter k is the maximum number of hops along the association path.

Define the set of triples passing through k hops as:

$$S^k = \{(h,r,t)|(h,r,t) \in G_2, h \in \mathcal{E}^{k-1}\}, k = 1,2,\cdots,H \quad (12)$$

2. Knowledge-aware attention embedding.

The target user's potential items of interest correspond to rich entities in the original graph, and the exploration of the user-end relational paths in the interaction graph alone is not sufficient to mine higher-order item attribute structures. To satisfy the derivation intent of Equation (6) to enrich the representation at the user end, the representation at the item end needs to enrich. Therefore, a knowledge-aware attention mechanism is introduced to aggregate the higher-order semantic and structural information of the user's potential item neighbourhood triples to provide a comprehensive description of the item through rich feature attributes.

In a knowledge graph, the triple structure indicates that multiple nodes may be connected to a significant node through multiple relationships. This means that these entities may be associated with a significant remote interest entity through a relational path. Considering these multiple entities as the starting point of the path as head entities and digging deeper along the association path, potential remote interest entities at the end of the path can be explored, and the potential interest entities at the end of the path are regarded as tail entities. Based on the above approach, modeling on the item side represents the tail entity as the remote similar attribute features of the item. The specific knowledge-aware attention embedding module is shown in Figure 4:

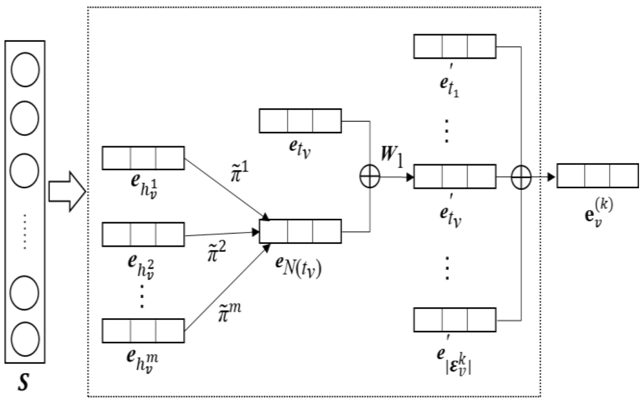


Figure 4. Knowledge perception attention embedding module.

Obtain multiple neighbouring head entities in the extended entity set that are oriented to a certain potential interest tail entity to form the set of neighbouring entities, defined as:

$$N(t_v) = \{h_v^m \mid h_v^m \in \mathcal{E}_v^{k-1} and (h_v^m, r_v^m, t_v) \in S_v^k\}, m = 1,2,\dots,M \quad (13)$$

Let there be m neighbouring head entities in the set of triples S_v^k at the kth level that are directed to the tail entities in the triples through the knowledge-aware graph attention, and these m head entities form the set of neighbourhood entities denoted as $N(t_v)$, with hyperparameters $M=|N(t_v)|$, and through the $N(t_v)$ to mine the neighbourhood representation of the tail entity. The process of aggregating the set of triples at the k^{th} layer is shown in Figure 4, where the embedding of the m neighbouring head entity $h_v^m \in N(t_v)$ is denoted as $e_{h_v^m}$, and the current m triples $(h_v^m, r_v^m, t_v) \in S_v^k$ are used to learn the fractional π^m in the graph-attentive network with the following equation:

$$\pi^m = LeakyReLU(W_2^\top [W_1 e_{h_v^m} || W_1 r_v^m || W_1 e_{t_v}]) \quad (14)$$

where LeakyReLU is used as the activation function, which enables the model to have a stronger nonlinear representation and better fit complex data distributions compared to the traditional ReLU. W_1 and W_2 are the weight matrix and the weight vector, respectively, which can be determined by parameter learning. In the propagation iteration $e_{h_v^m}$, r_v^m, e_{t_v} denote the d-dimensional embeddings vector representations of the head entity h_v^m, the relation r_v^m, and the tail entity t_v, respectively, and the symbol || refers to the connection operation between them.

The learning score π^m is normalized to obtain the corresponding weights $\tilde{\pi}^m$. The embedding $e_{h_v^m}$ of the neighbouring head entities are weighted by π^m and integrated to construct the neighbourhood feature representation $e_{N(t_v)}$ of the specified entity. The formula of the above process is represented as follows:

$$\tilde{\pi}^m = \frac{exp(\pi^m)}{\sum_{h_v^{m'} \in N(t_v)} exp(\pi^{m'})} \quad (15)$$

$$e_{N(t_v)} = \sum_{m=1}^{M} \tilde{\pi}^m e_{h_v^m} \quad (16)$$

The obtained neighbourhood representation $e_{N(t_v)}$ is combined with the embedded representation $e_{N(t_v)}$ of the potential interest-tailed entity t_v through knowledge perception to generate a higher-order representation e'_{t_v} of t_v, defined as follows:

$$e'_{t_v} = \sigma(W_1(e_{t_v} + e_{N(t_v)})) \quad (17)$$

where σ represents the sigmoid activation function and W_1 is the weight matrix.

3. Information dissemination aggregation

The above step is the processing flow for the neighbourhood triples of the k^{th} layer, where the higher-order embedded representations of the tail entity e'_{t_v} of different layers are accumulated to obtain the representation vector of the item v at the H^{th} layer by performing the same knowledge-aware operation on the triples S_v^k of the extended entity set ε_v^k in different propagation layers:

$$e_v^{(k)} = \sum_{t_v \in \varepsilon_v^k}^{|S_v^k|} e'_{t_v}, k = 1,2,\ldots,H \tag{18}$$

where $|S_v^k|$ is the number of triples in the set S_v^k.

The aggregated vectors of the final items are obtained by aggregating the representation vectors at each propagation level:

$$e_v = e_v^{(0)} + \cdots + e_v^{(H)} \tag{19}$$

3.6. UIKGAN Model Predictions

The aggregation vector of the user is obtained through Equation (10), and the aggregation vector of the item is obtained through Equation (19), and the aggregation vector of both ends is used as an input to predict the likelihood of the user u clicking on the un-interacted item v to interact with it through the MLP, with the following formula:

$$\hat{y}_{uv} = \sigma(F({e_u}^T e_v)) \tag{20}$$

where σ is a nonlinear sigmoid activation function and the prediction function F is expressed as an MLP.

The BPR loss function is utilized for training in the loss function :

$$\mathcal{L}_{BPR} = \sum_{(u,p,q)\in O} - ln\sigma(\hat{y}(u,p) - \hat{y}(u,q)) \tag{21}$$

where a negative sampling strategy $O=\{(u, p, q))|(u, p)\in R^+,(u, q)\in R^-\}$ is used, R^+ denotes the traversed observed user-item interactions, which are considered as positive case samples, and R^- denotes the pairs of interacting items that are not traversed and observed, which are considered as negative case samples. By combining the above module of independence loss modelling in Equation (3), the model loss function in this paper is defined as follows:

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_{TEN} + \lambda_2 \|\Theta\|_2^2 \tag{22}$$

where $\|\Theta\|_2^2$ is the $L2$ regularization term, Θ is the set of parameters of the model, and λ_1 and λ_2 are the equilibrium hyperparameters.

4. Experiments and Analysis of Results

4.1. Experimental Dataset

In order to better evaluate the performance of the model UIKGAN in this paper, 3 datasets are utilized: the book dataset (Book-Crossing) [14], the movie dataset (MovieLens-10M) [16], and the music dataset (Last.FM) [15], to ensure the consistency of the datasets in the comparison experiments. Each dataset was pre-processed before the experiment.

Table 2 shows the data details of the 3 experimental datasets:

Table 2. Dataset of the UIKGAN model experiment.

Dataset	Book-crossing	Movielens-10m	Last.Fm
Number of users	19,676	69,879	1,872
Number of items	20,003	10,601	3,846
Number of interactions	172,576	9,992,830	42,346
Number of entities	25,787	181,869	9,366
Number of relationships	18	51	60
Number of triads	60,787	95,580	15,518

4.2. Experimental Comparison Model and Parameter Settings

1. Comparison of baseline models.

In order to verify the effectiveness and performance of the proposed model UIKGAN in this paper, 6 other mainstream and representative recommendation models and the DNGAKG model in our previous work [7] (in press) are selected to carry out experimental comparisons. These 7 recommendation models all utilize a hybrid propagation strategy, which integrate the advantages of the graph embedding strategy and the path-based approach. The hyperparameters in the comparison baseline are set according to the optimal experimental parameters in the respective original papers:

- KNI [11]: this is an end-to-end neighbourhood interaction model for knowledge-enhanced recommendation. The algorithm proposes a Neighbourhood Interaction Model (NI), thus uniquely capturing the neighbour relationship between the user end and the item end. In NI, additional Knowledge Graphs (KGs) are added and combined with Graph Neural Networks (GNN) for accurate recommendation through a module of knowledge-enhanced neighbourhood interaction.
- KGAT [21]: this model fuses historical user interaction information with the knowledge graph to construct a CKG, builds an attention-aware representation propagation layer, and employs a knowledge graph attention network to explicitly model the higher-order structural information in the graph in an end-to-end manner.
- KGNN-LS [19]: this is a more advanced recommendation model based on the propagation method, which converts heterogeneous graph information into the form of user-weighted graphs, and employs label smoothing to propagate the user's labelling information in GNN. This model effectively improves the recommendation performance while making the model with good generalization ability.
- KGIN [22]: this model makes recommendation more personalized by revealing user tendencies behind knowledge graph interaction information. Under the GNN framework, it refines the user's preference division and captures deeper user collaboration information in the graph using relational paths, thus enriching the description of user behavioural

patterns.

- CKGAT [25]: this is an improved upgrade of the collaborative knowledge-aware attention network model CKAN [23], which improves the original model's defect of not distinguishing important relationships among different entities in the path ripple set, extracts the topological nearest neighbour structure in the multi-hop ripple set through knowledge-aware techniques, and finally embeds the more refined ripple set into the aggregation through an attention aggregator.
- MI-KGNN [24]: this model characterizes the similarity between users and items through information propagation and aggregation in the knowledge graph, fully explores the multidimensional interactions between nodes during information propagation, optimizes the update direction of node representations, and optimizes the weights of information propagation by using the dual attention mechanism.
- DNGAKG [7] (in press): this model traverses the knowledge graph of user-item interactions from both the user end and the item end, fully mines the feature information of users and items in the graph, and effectively improves the recommendation performance and interpretability.

2. Hyperparameters setting

In the UIKGAN model, for different datasets, the data is randomly split into training, validation, and test sets in a 6:2:2 ratio. Negative examples are randomly sampled as the control, while the remaining data serves as positive examples. The final hyperparameter values are set by optimizing the AUC value on the validation set. The initialization of parameters is set by Xavier initializer, the training of the model is optimized by Adamax optimizer, and the model is accelerated by local GPU.

The hyperparameters on the user end and item end are set as follows:

1. At the user end: the number of user intention tendency factors I is selected in the set {2, 4, 8, 16, 32}; meanwhile, in the relational path attention aggregation module, the path propagation depth L is selected in the set {1, 2, 3, 4, 5}.
2. At the item end: the number of propagation aggregation layers H in the knowledge-aware attention-based embedding module is selected in the set {1, 2, 3, 4, 5}, and for the aggregation process the number of neighbourhood triplet samples $|S_v^k|$ is selected in the set {4, 8, 16, 32, 64, 128}. The embedding dimension d is adjusted between {8, 16, 32, 64, 128, 256}; the intention-independent modelling parameter λ_1 and the L2 regularization parameter λ_2 are adjusted between {10-5, 10-4, 10-3, 10-2}, and the learning rate ρ is adjusted between {10-4, 10-3, 10-2, 10-1}.

The batch size in model training is uniformly set to 1024.

The optimal hyperparameter settings for the UIKGAN model are shown in Table 3:

Table 3. Details of hyperparameter settings for the UIKGAN model.

Hyperparamete settings	Book-crossing	MovieLens-10M	Last.FM
I	16	16	32
L	4	4	3
H	3	3	4
$\lvert S_v^k\rvert$	64	64	64
d	64	64	64
λ_1	10-5	10-5	10-4
λ_2	10-5	10-5	10-5
ρ	10-4	10-4	10-4
Batch size	1024	1024	1024

4.3. Experimental Evaluation Metrics

When evaluating the performance of recommendation algorithms, two recommendation scenarios are considered: Click Through Rate (CTR) prediction and Top-K (performance evaluation of the prediction samples ranked in the top-K) recommendation. In the prediction experiments under the CTR scenario, Area Under the Curve (AUC) and F1 (F1-Score) are used as the evaluation metrics; Recall as the evaluation metric in the Top-K recommendation scenario.

The AUC is used to measure the performance of a model by calculating the area under the Receiver Operating Characteristic (ROC) curve. The ROC curve plots the true positive rate against the false positive rate, with the horizontal axis ranging from 0 to 1. Consequently, the AUC value also falls within the range of 0 to 1, where a higher value indicates better model performance. The AUC can be expressed in terms of the integral of the ROC curve as follows:

$$AUC = \int_0^1 f(ROC)dx \tag{23}$$

Where $f(ROC)$ is a function of the ROC curve.

F_1 is an evaluation metric that combines precision and recall, with the following equation:

$$AUC = \int_0^1 f(ROC)dx \tag{24}$$

where P is the precision rate, R is the recall rate, and F_1 is the reconciled average of the precision rate and the recall rate, which is used to comprehensively assess the performance of the model in classifying positive and negative samples, where both P and R are numbers with values ranging from 0 to 1, and F_1 reflects the comprehensive expectation information.

To discriminate whether the predicted and true values of a sample are the same, the following types exist:

1. True Positive (TP): the number of positive examples predicted as positive.
2. False Negative (FN): the number of positive examples predicted as negative.
3. False Positive (FP): the number of negative examples predicted as positive examples.

4. True Negative (TN): the number of negative examples predicted as negative.

Recall reflects the proportion of true positive results among all actual positive instances, the formula is:

$$Recall = \frac{TP}{TP + FN} \tag{25}$$

Classifying positive examples as negative will reduce the recall rate. *Recall* measures a model's ability to identify all true positive instances. This metric reflects the model's effectiveness in uncovering genuine positive cases and indicates how well the recommendation model can identify the user's potential interests.

4.4. Experimental Results and Analysis

1. Comparative analysis of experiments in CTR scenarios.

The experimental results for the CTR prediction, comparing the UIKGAN model with 7 other baseline models, are shown in Table 4. As can be seen from Table 4, the evaluation metrics of the UIKGAN model on all 3 datasets are better than the baseline of the comparison, which effectively improves the prediction performance in CTR scenario. Compared with the experimentally obtained optimal metrics of the comparison model, AUC and F1 are improved by 1.99% and 0.59% on the Book-Crossing dataset, 0.20% and 0.53% on the MovieLens-10M dataset, and 0.71% and 1.03% on the Last.FM dataset, respectively. It can be found that the performance improvement of our model UIKGAN is better on the Book-Crossing dataset and the Last.FM dataset than on the MovieLens-10M dataset. The reason for this is that the MovieLens-10M dataset is the densest compared to the other two datasets with huge interaction data information. However this denseness may instead affect the propagation effect of the model in the graph, and a small performance gain is realized in this scenario compared to the DNGAKG model that has the best performance.

Table 4. Experimental results of UIKGAN in CTR scenario.

Model	Book-crossing		MovieLens-10M		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
KNI	0.690	0.627	0.971	0.919	0.807	0.711
KGAT	0.733	0.655	0.975	0.929	0.829	0.747
KGNN-LS	0.689	0.636	0.978	0.926	0.811	0.726
KGIN	0.755	0.675	0.977	0.934	0.842	0.761
CKGAT	0.753	0.680	0.979	0.928	0.851	0.768
MI-KGNN	0.745	0.671	0.980	0.939	0.848	0.770
DNGAKG	0.751	0.679	0.983	0.942	0.818	0.736
UIKGAN	0.770	0.684	0.985	0.947	0.857	0.778

From the experimental results in Table 4, it can be seen that the performance of each baseline on MovieLens-10M dataset is good, and the gap between the evaluation scores on the other two datasets is relatively small, thanks to the fact that the baseline models in this paper's experiments all use mainstream propagation-based recommendation methods, and the original models namely have good recommendation performance and prediction accuracy.

The KGIN model focuses on building association paths in the graph. Our model refers to the modelling concept of this model on the user end, and models the user-item history interactions at a fine-grained level. Compared with the KGIN model, the AUC are improved by 1.99%, 0.82% and 1.78%, and the F1 evaluation metrics are improved by 1.33%, 1.39% and 2.23% in the experimental scenarios of the above 3 datasets, respectively. The reason is that, although the KGIN model reveals the intention behind user-item interactions through the user's perspective, and improves the interpretability of the model by coupling the intention relation with the knowledge graph relation in the data, it does not explore the complex attention module in the module of integrating remote semantics by using the relational paths, and lacks the knowledge perception employed by the UIKGAN model attention module. As a result, even with the construction of multi-hop relational paths, the KGIN model still loses potential semantic information during the information dissemination process, resulting in a slightly inferior performance to the model in this paper.

Comparing with the CKGAT model referenced on the item end, AUC are improved by 2.26%, 0.61% and 0.71%, and the F1 are improved by 0.59%, 2.05%, and 1.30%, respectively, in the experimental scenarios of the 3 datasets mentioned above. Comprehensively analyzing the experimental results, the model's performance in the experimental scenario is slightly better than KGIN, and the performance metric of UIKGAN has a smaller improvement compared with it. A comprehensive analysis of the experimental results shows that the UIKGAN model performs slightly better than KGIN in the experimental scenario, with only a marginal improvement in performance metrics. The reason for this can be attributed to the CKGAT model, which takes into account the complex relationships between entities in multiple-hop propagation layers. By utilizing attention networks to aggregate higher-order neighborhood information associated with the entities in the multi-hop ripple concentration graph, CKGAT effectively captures users' historical interest features on a broader scale. Incorporating this idea into the item end of the model in this paper, the knowledge-aware module is applied to capture similar attribute features of items remotely in the original graph, and the potential features of items in different layers are aggregated in the propagation, which enriches the descriptions of the user's potential items of interest, and thus makes the final prediction more accurate.

Comparing with our previously proposed DNGAKG model under the same experimental scenarios, the AUC are improved by 2.53%, 0.20% and 4.77%, and the F1 are improved by 0.74%, 0.53% and 5.71%, respectively. Comprehensive analysis reveals that the UIKGAN model outperforms the DNGAKG model, with a significant performance improvement on the sparse Last.FM dataset. The Last.FM dataset contains fewer interaction data and triples compared to the other 2 datasets, indicating the effectiveness of the proposed model in associative path modeling. UIKGAN is capable of effectively mining structural information from sparse datasets.

2. Comparative analysis of experiments in Top-K recommendation scenarios.

In Top-K recommendation scenario, for each user in the test set, the top K items in terms of predicted CTR are selected. The value of K is set to {5, 10, 20, 50, 100}. The experimental results are shown in Figures 5, 6, and 7, where the horizontal coordinate indicates the value of K and the vertical coordinate indicates the performance score of *Recall*.

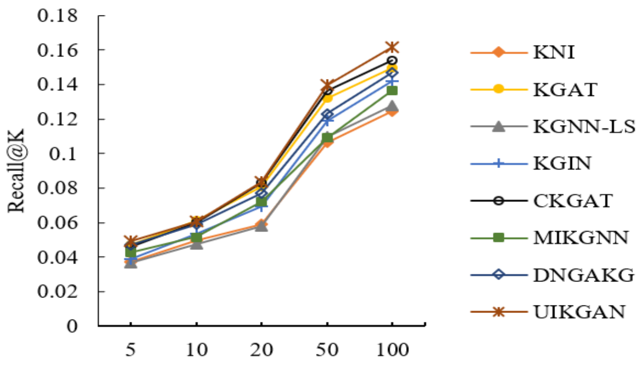


Figure 5. UIKGAN model recall results on book-crossing dataset.

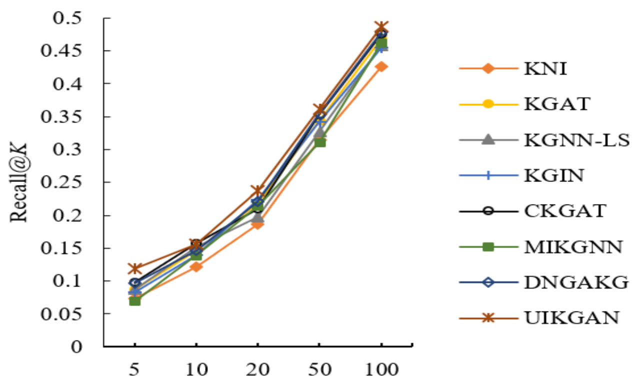


Figure 6. UIKGAN model recall results on MovieLens-10M dataset.

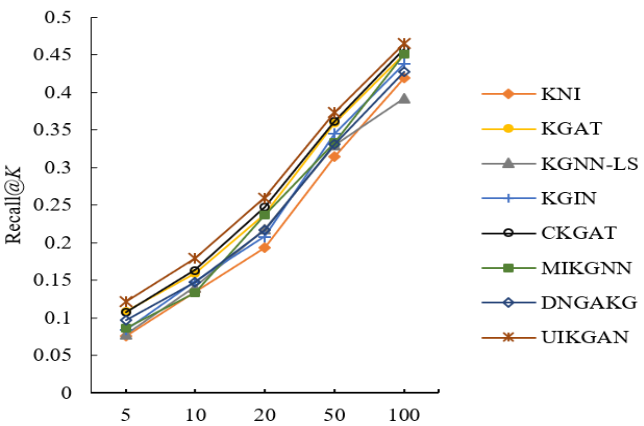


Figure 7. UIKGAN model recall results on Last.FM dataset.

From Figures 5, 6, and 7, it can be seen that on the 3 datasets, the Recall values of UIKGAN model and the compared baselines rise to varying degrees as the value of K increases, and the overall Recall of UIKGAN is better than that of the other models in this scenario; in the MovieLens-10M dataset, when the value of K is from 20 to 50, the Recall value improves the most. In the MovieLens-10M dataset, the Recall value improves most when the K value is from 20 to 50. When the K value is 20, the Recall of UIKGAN is 12.64% and 11.48% higher than the latest models MI-KGNN and DNGAKG, and 6.21% and 13.34% higher than the reference models KGIN and CKGAT, respectively, which is a significant performance enhancement. The Recall results of CKGAT are only second to that of our model UIKGAN, which again verifies that the Knowledge Awareness Attention Module has better overall recall than other models in this scenario. MI-KGNN model has an average performance, which indicates that the overuse of attention mechanism is not necessarily applicable to the actual recommendation scenarios. KNI model has a good design concept, i.e., to improve the recommendation performance through neighbourhood interaction and knowledge enhancement, but the recommendation performance in this scenario is poor, which indicates that KNI has poor interpretability and is not suitable for dealing with graph data with complex interaction information.

3. Ablation experiment.

The purpose of this experiment is to investigate whether UIKGAN model can improve the accuracy of prediction and the comprehensive model performance by modelling the independence of user's intention on the user's end and by incorporating the MLP into the prediction framework to calculate the potential interaction probability. The performance comparison results of the ablation experiments on the 3 datasets are shown in Table 5.

Table 5. Ablation experiments of UIKGAN model.

Model	Book-crossing		MovieLens-10M		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
UIKGAN	0.770	0.684	0.985	0.947	0.857	0.778
UIKGAN-d	0.749	0.672	0.974	0.940	0.847	0.761
UIKGAN-m	0.766	0.676	0.979	0.939	0.859	0.777

UIKGAN-d indicates that user independence modelling is not introduced on the user end. The experimental comparison in Table 5 shows that on the 3 datasets, the AUC and F1 values of UIKGAN are better than UIKGAN-d, therefore, the intent independence modelling is an integral part of the user intent preference module, which optimizes the loss function and effectively improves the performance of the recommendation at the same time.

UIKGAN-m indicates that an MLP is not used in the prediction module. UIKGAN model compares with the UIKGAN model in terms of AUC and F1 values by

0.52% and 1.18% on the Book-Crossing dataset, 0.61% and 0.85% on the MovieLens-10M dataset, and 0.61% and 0.85% on the Last.FM dataset. An anomaly occurs on the Last.FM dataset, where removing the MLP instead performs better. It indicates that overuse of neural networks for aggregation on sparse datasets may have side effects.

4. Hyperparametric sensitivity analysis.

The hyperparameters of the UIKGAN model are adjusted in the experiment, including the embedding dimension *d*, the number of user intent preference factors *I*, the propagation depth of the relationship path *L*, the number of propagation aggregation layers *H*, and the number of neighbourhood ternary samples $|S_v^k|$, in order to study the impact of the values of each parameter on the recommendation performance of the model UIKGAN in this chapter, and the specific experimental results are as follows:

Figures 8 and 9 show the effect of the embedding dimension *d* of entities and relations on the model performance, with the range of *d* in {8,16,32,64,128,256}.

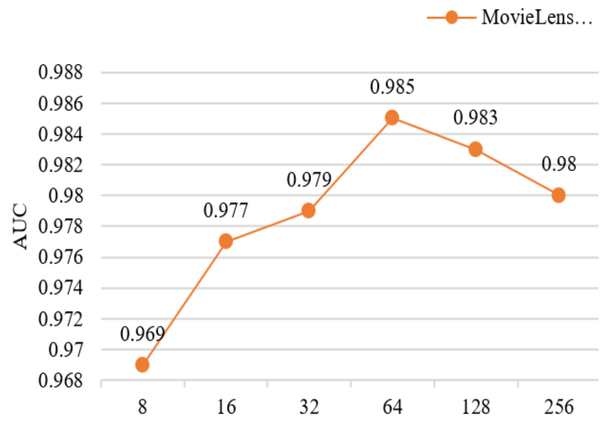


Figure 8. AUC results of UIKGAN model on MovieLens-10M with different embedding dimensions.

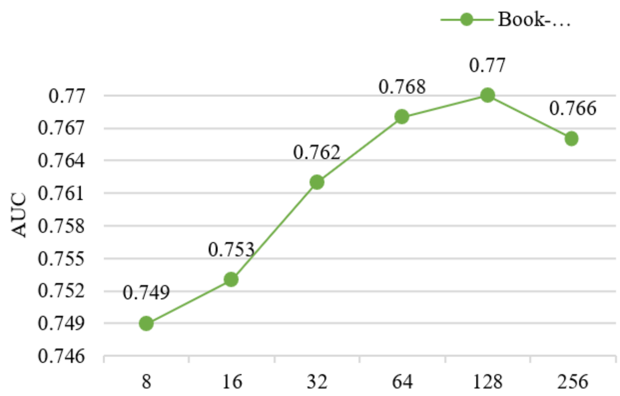


Figure 9. AUC results of UIKGAN model on book-crossing with different embedding dimensions.

Experiments are conducted on the dense MovieLens-10M dataset and Book-Crossing dataset. As can be seen from Figures 8 and 9, within a certain range, the model performance keeps improving as the value of *d* increases, which is due to the fact that higher dimensional embeddings can better encode the semantic information between entities and relations. However, when the embedding dimension d exceeds a specific value, the model performance tends to decrease, which may be due to the fact that higher dimensions cause overlearning of the model, and the model overfits to the noise or sample-specific information in the training data, which subsequently triggers a decline in the performance on the test set. To avoid this, suitable regularization techniques or increasing the diversity of the training dataset are required to improve the generalization ability of the model.

Table 6. Effect of the number of user intention tendency factor I on AUC value in UIKGAN.

I	Book-Crossing	MovieLens-10M	Last.FM
2	0.737	0.941	0.814
4	0.756	0.967	0.832
8	0.764	0.975	0.851
16	0.770	0.985	0.854
32	0.767	0.980	0.857

Table 6 shows the effect of the number of user intent tendency factor *I* on the model performance. When *I*=2, the model performs poorly on all three datasets, indicating that effective modelling of user intent is decisive for the quality of recommendation results in UIKGAN. On Book-Crossing and MovieLens-10M datasets, the best AUC value is obtained when *I*=16, and the performance decreases instead when it is increased to 32. The reason is that the huge MovieLens-10M dataset is too rich in interaction information and corresponding graph relationships, and the excessive segmentation of the *I* module introduces noise interference instead. Comparison of the experimental results shows that adjusting the number of user intention tendency factors in a suitable range can increase the granularity of modelling user collaboration information to the intention level, describe the user's behavioural patterns at a finer granularity, and improve the accuracy of model prediction.

Table 7. Impact of relational path propagation depth L on AUC values in UIKGAN.

L	Book-crossing	MovieLens-10M	Last.FM
1	0.742	0.970	0.839
2	0.754	0.973	0.848
3	0.771	0.979	0.858
4	0.768	0.985	0.855
5	0.757	0.983	0.850

Table 7 shows the effect of relationship path propagation depth *L* on model performance. Relational paths are modelled on the user end, following the construction idea of the UIKGAN model, even if the user's potentially preferred entities and their semantic information are far away, they can be mined through the links of the relational propagation paths. In the actual experimental scenario, the propagation distance of the path needs to be specifically analyzed. The experimental results show that the optimal propagation depth *L* is 3 on both Book-Crossing and Last.FM datasets, which shows that in specific applications, the propagation depth is not as large as better. On the MovieLens-10M dataset, the best performance is achieved when the value of L is taken as 4. This stems from the fact that the number of triples in this dataset is more than that in the other two datasets, and thus deeper relational path extensions are able to traverse the semantic and structural information of the triples well.

Table 8. Effect of the number of propagation aggregation layers H on AUC values in UIKGAN.

H	Book-Crossing	MovieLens-10M	Last.FM
1	0.759	0.976	0.843
2	0.763	0.983	0.851
3	0.770	0.986	0.848
4	0.768	0.981	0.857
5	0.754	0.979	0.853

Table 8 shows the effect of the number of propagation aggregation layers H on the model performance. The propagation aggregation layer is modelled on the item end and is used to process the triplet data in the item knowledge graph, thus enriching the vector representation of the items. The best performance is achieved when H is 3 for the first two datasets, and the values of H between {2, 3, 4} have good results. Compared with the other parameters, the values of the number of aggregation layers have a slightly smaller impact on the model. Therefore, when dealing with large datasets, choosing the appropriate number of aggregation layers can maximize the effect of the knowledge-aware network.

Table 9. Influence of neighbourhood triplet sampling number $|S_v^k|$ on AUC values in UIKGAN.

$\lvert S_v^k \rvert$	Book-Crossing	MovieLens-10M	Last.FM
4	0.757	0.963	0.836
8	0.761	0.971	0.847
16	0.765	0.977	0.852
32	0.768	0.982	0.854
64	0.770	0.985	0.858
128	0.753	0.979	0.850

Table 9 shows the effect of the neighbourhood triplet sampling number on the model performance. The number of triplet samples here corresponds to the number of triplet entities in each layer of aggregation in the experiments in Table 8. The value of AUC increases gradually as the number of neighbouring triplet increases. A sampling number of 128 is an overly large and unreasonable choice, as it may introduce noise. In contrast, a sampling number of 64 is a more appropriate option. In knowledge graphs, triplet exhibit excellent relational representation and structuring properties. When used as auxiliary information, they can enrich the description of item attributes, thus effectively improving the recommendation effect.

5. Experiments in analyzing sparsity problems.

Knowledge graph as auxiliary information plays a significant role in alleviating the sparsity problem of recommendation models. In order to deeply study the model performance in sparsity scenarios, the MovieLens-10M dataset is selected as the experimental object. During the comparison experiments, the sizes of validation and test sets are kept unchanged, and the size of the training set is gradually adjusted, which is set to 100%, 80%, 60%, 40% and 20% of the original training set, respectively. Figure 10 demonstrate the performance of AUC evaluation of each baseline model under different training set ratios.

As seen in Figure 10, the KGNN-LS model has the largest decrease in AUC performance and the UIKGAN model has the most moderate decrease in AUC metrics. Equivalent to the training metrics on the initial 100% training set, the AUC metrics of the KNI, KGAT, KGNN-LS, KGIN, CKGAT, MIKGNN, DNGAKG, and UIKGAN models decreased by 5.36%, 4.21%, 6.24%, 3.89%, 3.87%, 3.92%, 5.09%, and 4.06% respectively. In general, the UIKGAN model in this paper maintains good performance with CKGAT and MIKGNN models on sparse datasets. The enhancement of recommendation performance by the propagation-based approach is again verified.

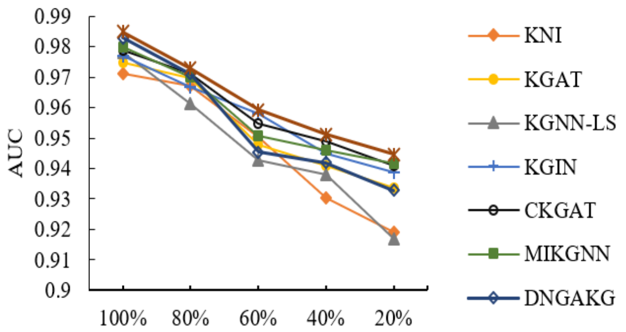


Figure 10. AUC values under different ratio training sets for the baseline model.

5. Conclusions and Future Work

This paper proposes UIKGAN, a dual-end recommendation model that integrates user intent modeling and knowledge-aware graph attention. On the user side, a user intent preference module captures fine-grained collaborative information by modeling historical interactions along different propagation paths. On the item side, knowledge-aware graph attention aggregates higher-order triples into item vectors, which are fused with user intent representations to predict preferences.

Comprehensive experiments, including comparisons with 7 benchmarks, ablation studies, and hyperparameter analysis, demonstrate that UIKGAN exceeds in CTR prediction and Top-K recommendation. The model outperforms baselines in both performance and interpretability, which demonstrates its effectiveness for sparse datasets and practical value in recommendation systems.

The model UIKGAN proposed in this paper shows certain advantages in practical application scenarios. Combined with the latest research trends in the field of recommendation systems, there is still a potential for improvement to be explored. Future research can further deepen the following aspects:

1. In the modeling of user intention tendency, more research is needed to optimize and improve the division of intention and the selection of quantity; and the algorithm may have certain challenges when dealing with large-scale datasets, and the applicability of the model needs to be further verified by datasets from other domains.
2. Most of the contemporary models are based on static modeling with fixed temporal data, given that the user's preference in real-world scenarios may change over time or social relationships, resulting in different interaction behaviors. Therefore, knowledge graph information can be combined with temporal information to dynamically model the given model.
3. The dataset processed in this study is based on the construction of a knowledge graph based on the encyclopedia class, and future research can consider integrating additional information sources such as social networks to further enhance the modelling representation of users and items. The way in which path information and relational dependencies are combined in the model can be further investigated in the future to better utilize this information for recommendation purposes. Integration of knowledge graphs from more domains can be considered as auxiliary information to provide in-depth semantic understanding and knowledge contextual support for the model.

Acknowledgment

The work is supported by Colleges Nature Science Research Key Project of Anhui Province (No.KJ2021A1157), Master's Supervisor Research Funding Project (DSJF008), Anhui Province Quality Engineering (No.2021yljc063, 2020kcszjxtd37), Introducing Talent Research Initiation Project of Anhui Xinhua University (2015kyqd002).

References

[1] Abdi M., Okeyo G., and Mwangi R., "Improved Collaborative Filtering Recommender System Based on Hybrid Similarity Measures," *The International Arab Journal of Information Technology*, vol. 22, no. 1, pp. 99-115, 2025. https://doi.org/10.34028/iajit/22/1/8

[2] Cao Y., Shang S., Wang J., and Zhang W., "Explainable Session-based Recommendation via Path Reasoning," *arXiv Preprint*, vol. arXiv:2403.00832v1, pp. 1-13, 2024. https://arxiv.org/abs/2403.00832

[3] Cao Y., Wang X., He X., Hu Z., and Chua T., "Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preference," *in Proceedings of the 28th International World Wide Web Conference*, San Francisco, pp. 151-161, 2019. https://doi.org/10.1145/3308558.3313705

[4] Chen J. and Zhang W., "Review of Point of Interest Recommendation Systems in Location-Based Social Networks," *Journal of Frontiers of Computer Science and Technology*, vol. 16, no. 7, pp. 1462-1478, 2022. https://doi.org/10.3778/j.issn.1673-9418.2112037

[5] Hamilton W., Ying R., and Leskovec J., "Inductive Representation Learning on Large Graphs," *in Proceedings of the 31st International Conference on Neural Information Processing Systems*, California, pp. 1025-1035, 2017. https://dl.acm.org/doi/10.5555/3294771.3294869

[6] Ji W., Wang H., Su G., and Liu L., "Review of Recommendation Methods Based on Association Rules Algorithm," *Computer Engineering and Applications*, vol. 56, no. 22, pp. 33-41, 2020. http://cea.ceaj.org/CN/10.3778/j.issn.1002-8331.2006-0158

[7] Ji Z. and Lv T., "A Dual End Neighbour Recommendation Algorithm Integrating Graph Attention and Knowledge Graph Convolutional Networks," *Journal of Huaibei Normal University (Natural Science Edition)*, vol. 45, no. 3, pp. 50-58, 2024. https://fmsb.cbpt.cnki.net/WKE2/WebPublication/paperDigest.aspx?paperID=f5d788e1-b122-45c0-b52e-b545a5d9c9a2#

[8] Kipf T. and Welling M., "Semi-Supervised Classification with Graph Convolutional Networks," *arXiv Preprint*, vol. arXiv:1609.02907v4, pp. 1-14, 2017. https://10.48550/arXiv.1609.02907

[9] Lin J., Chen S., and Wang J., "Graph Neural Networks with Dynamic and Static Representations for Social Recommendation," *in Proceedings of the 27th International Conference on Database Systems for Advanced Applications*, Hyderabad, pp. 264-271, 2022. https://doi.org/10.1007/978-3-031-00126-0_1

[10] Lyu Z., Wu Y., Lai J., Yang M., and Li C., and Zhou W., "Knowledge Enhanced Graph Neural Networks for Explainable Recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4954-4968, 2023. DOI: 10.1109/TKDE.2022.3142260

[11] Qu Y., Bai T., Zhang W., Nie J., and Tang J., "An End-to-End Neighbourhood-based Interaction Model for Knowledge-Enhanced Recommendation," *in Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, Anchorage, pp. 1-9, 2019. https://doi.org/10.1145/3326937.3341257

[12] Sha X., Sun Z., and Zhang J., "Hierarchical Attentive Knowledge Graph Embedding for Personalized Recommendation," *Electronic Commerce Research and Applications*, vol. 48, no. 1, pp. 1-14, 2021. https://doi.org/10.1016/j.elerap.2021.101071

[13] Tang X., Wang T., Yang H., and Song H., "AKUPM: Attention-Enhanced Knowledge-Aware User Preference Model for Recommendation," *in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Anchorage, pp. 1891-1899, 2019. https://doi.org/10.1145/3292500.3330705

[14] The Book-Crossing Dataset, http://www2.informatik.uni-freiburg.de/~cziegler/BX/, Last Visited, 2025

[15] The Last.FM Dataset, https://www.last.fm/api, Last Visited, 2025

[16] The MovieLens 10M Dataset, https://grouplens.org/datasets/movielens/10m/, Last Visited, 2025

[17] Tian X. and Chen H., "Survey on Applications of Knowledge Graph Embedding in Recommendation Tasks," *Journal of Frontiers of Computer Science and Technology*, vol. 16, no. 8, pp. 1681-1705, 2022. http://fcst.ceaj.org/CN/10.3778/j.issn.1673-9418.2112070

[18] Wang H., Zhang F., Wang J., Miao Z., Li W., Xie X., and Guo M., "RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems," *in Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Torino, pp. 417-426, 2018. https://doi.org/10.1145/3269206.3271739

[19] Wang H., Zhang F., Zhang M., Leskovec J., Zhao M., Li W., and Wang Z., "Knowledge-Aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems," *in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Anchorage, pp. 968-977, 2019. https://doi.org/10.1145/3292500.3330836

[20] Wang H., Zhao M., Xie X., Li W., and Guo M., "Knowledge Graph Convolutional Networks for Recommender Systems," *in Proceedings of the World Wide Web Conference*, San Francisco, pp. 3307-3313, 2019. https://doi.org/10.1145/3308558.331341

[21] Wang X., He X., Cao Y., Liu M., and Chua T., "KGAT: Knowledge Graph Attention Network for Recommendation," *in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Anchorage, pp. 950-958, 2019. https://doi.org/10.1145/3292500.3330989

[22] Wang X., Huang T., Wang D., Yuan Y., Liu Z., He X., and Chua T., "Learning Intents behind Interactions with Knowledge Graph for Recommendation," *in Proceedings of the Web Conference*, Ljubljana, pp. 878-887, 2021. https://doi.org/10.1145/3442381.3450133

[23] Wang Z., Lin G., Tan H., Chen Q., and Liu X., "CKAN: Collaborative Knowledge-Aware Attentive Network for Recommender Systems," *in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Xi'an, pp. 219-228, 2020. https://doi.org/10.1145/3397271.3401141

[24] Wang Z., Wang Z., Li X., Yu Z., Guo B., Chen L., "Exploring Multi-Dimension User-Item Interactions with Attentional Knowledge Graph Neural Networks for Recommendation," *IEEE Transactions on Big Data*, vol. 9, no. 1, pp. 212-226, 2022. DOI:10.1109/TBDATA.2022.3154778

[25] Xu Z., Liu H., Li J., Zhang Q., and Tang Y., "CKGAT: Collaborative Knowledge-Aware Graph Attention Network for Top-N Recommendation," *Applied Sciences*, vol. 12, no. 3, pp. 1-23, 2022. https://doi.org/10.3390/app12031669

[26] Yang C., Chen X., Wang C., and Liu T., "Recommendation Strategy Based on Users' Preferences for Fine-Grained Attributes," *Data Analysis and Knowledge Discovery*, vol. 5, no. 10, pp. 94-102, 2021. DOI: 10.11925/infotech.2096-3467.2021.0291

[27] Yin Y., Zhu X., Wang W., Zhang Y., Wang P., Fan Y., and Guo J., "HEC-GCN: Hypergraph Enhanced Cascading Graph Convolution Network for Multi-Behavior Recommendation," *arXiv Preprint*, vol. arXiv:2412.14476v1, pp. 1-12, 2024. https://arxiv.org/abs/2412.14476

[28] Zhang M., Zhang X., Liu S., Tian H., and Yang Q., "Review of Recommendation Systems Using Knowledge Graph," *Computer Engineering and Applications*, vol. 59, no. 4, pp. 30-42, 2023. http://cea.ceaj.org/CN/10.3778/j.issn.1002-8331.2209-0033

[29] Zhao Y., Liu L., Wang H., Han H., and Pei D., "Survey of Knowledge Graph Recommendation System Research," *Journal of Frontiers of Computer Science and Technology*, vol. 17, no. 4, pp. 771-791, 2023. https://doi.org/10.3778/j.issn.1673-9418.2205052

[30] Zou L., Xia L, Gu Y., Zhao X., Liu W., Huang J., and Yin D., "Neural Interactive Collaborative Filtering," *in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Xi'an, pp. 749-758, 2020. https://doi.org/10.1145/3397271.3401181

Zijie Ji is a Master's candidate, his research focuses on Recommendation Systems.

Teng Lv is a Professor, Ph.D., his research focuses on Recommendation Systems and Data Management. He is the Corresponding author of this paper.

Yi Yu is a Master's candidate, his Research Focuses on Recommendation Systems.