Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.046560

ARTICLE

Recommendation Method for Contrastive Enhancement of Neighborhood Information

Hairong Wang, Beijing Zhou*, Lisi Zhang and He Ma

School of Computer Science and Engineering, North Minzu University, Ningxia, 750021, China

*Corresponding Author: Beijing Zhou. Email: 20217381@stu.nmu.edu.cn

Received: 07 October 2023 Accepted: 14 November 2023 Published: 30 January 2024

ABSTRACT

Knowledge graph can assist in improving recommendation performance and is widely applied in various personalized recommendation domains. However, existing knowledge-aware recommendation methods face challenges such as weak user-item interaction supervisory signals and noise in the knowledge graph. To tackle these issues, this paper proposes a neighbor information contrast-enhanced recommendation method by adding subtle noise to construct contrast views and employing contrastive learning to strengthen supervisory signals and reduce knowledge noise. Specifically, first, this paper adopts heterogeneous propagation and knowledge-aware attention networks to obtain multi-order neighbor embedding of users and items, mining the high-order neighbor information of users and items. Next, in the neighbor information, this paper introduces weak noise following a uniform distribution to construct neighbor contrast views, effectively reducing the time overhead of view construction. This paper then performs contrastive learning between neighbor views to promote the uniformity of view information, adjusting the neighbor structure, and achieving the goal of reducing the knowledge noise in the knowledge graph. Finally, this paper introduces multi-task learning to mitigate the problem of weak supervisory signals. To validate the effectiveness of our method, experiments are conducted on the MovieLens-1M, MovieLens-20M, Book-Crossing, and Last-FM datasets. The results show that compared to the best baselines, our method shows significant improvements in AUC and F1.

KEYWORDS

Contrastive learning; knowledge graph; recommendation method

1 Introduction

Recommendation algorithms excel at extracting user-interest content from vast information repositories, tailoring it to individualized preferences. Early incarnations of these algorithms were largely anchored around collaborative filtering (CF) [1–3]. Such techniques are modeled based on user-item interaction histories to prognosticate user inclinations. However, they often confronted the twin challenges of sparse user-item interaction data and the cold start problem for nascent users. To circumvent these issues, researchers introduced auxiliary information sources [4,5]. These encompassed user profiles [6], item characteristics [7,8], social networks [9,10], and knowledge graphs [11,12]. Notably, knowledge graphs, as supplemental data, adeptly capture inter-item structural and semantic nuances, amplifying recommendation accuracy and augmenting the explicability of the recommendation outcomes.

Knowledge Graphs (KG) house a wealth of semantically interlinked knowledge, forging connections between entities and the users and items in recommendations. This is instrumental in unveiling latent relationships between users and items within the KG, thereby alleviating data sparsity and the cold start conundrum. In the current paradigm, knowledge graphs have ascended as a potent supplementary resource in recommendation strategies, witnessing widespread adoption.

Depending on their application in recommendation methods, techniques leveraging knowledge graphs can generally be categorized into those based on embeddings, path-centric approaches, and propagation-based strategies. Among these, propagation-centric methodologies emerge as one of the dominant paradigms, tapping into the structural and semantic assets of the knowledge graph. By employing the Graph Convolution Network (GCN) [13], these methods aggregate information from neighboring nodes and recursively propagate to uncover latent attributes and associations of users and items within the knowledge graph. Notable implementations of this paradigm include models like KGAT [14], KGIN [15], and CKAN [16]. While these propagation-based techniques exhibit prowess in assimilating information from neighbor nodes to enhance representations of users and items, consequently uplifting recommendation performance, they are not devoid of lingering challenges.

(1) Ignore the fact that there is a lot of noisy knowledge in KG. In the real world, KG contains a large amount of erroneous and irrelevant information, which is often unrelated to the recommendation task. Indiscriminately aggregating this knowledge can lead to suboptimal recommendation performance. As illustrated in Fig. 1, items i_1, i_2, and i_3 interact with user u_1. Among them, i_1 is associated with relationships r_1 and r_2, while i_2 and i_3 are linked with relationships r_2 and r_3. We can infer that user u_1 has preferences for a movie's lead actors and genres, but relationship r_1 acts as noise for user u_1. In real scenarios, KG usually contains a vast amount of noisy and redundant knowledge. Using GCN to aggregate neighbor information amplifies the noise in KG, resulting in subpar recommendation performance.

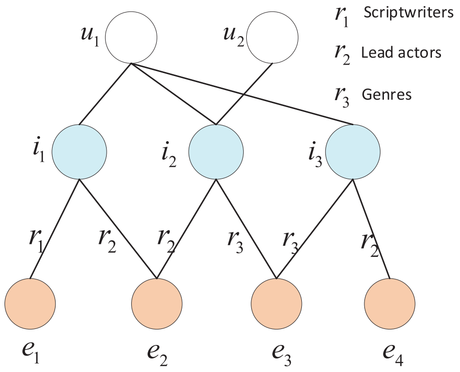


Figure 1: Example of movie recommendation scene

(2) Weak supervisory signals. Most knowledge-aware recommendation methods rely on user-item interactions as the supervisory signal. However, in real-world scenarios, the interaction history between users and items is extremely sparse. If we rely solely on the user-item interaction information as the supervisory signal, it can lead to the problem of weak supervisory signals, thereby affecting the recommendation results.

To address the problems of knowledge noise in KG and weak supervisory signals, this paper proposes a recommendation method called RMCEN (Recommendation Method for Contrastive Enhancement of Neighborhood Information). Specifically, this method first combines the knowledge graph and user-item interaction information through heterogeneous propagation to explicitly encode the collaborative signal between users and items and generate multi-order neighbor information. Next, this paper introduces a knowledge attention function to capture essential knowledge in neighbor information, reducing knowledge noise in neighborhood information. Concurrently, it involves constructing contrast views of neighbor information and employs contrastive learning to further diminish the knowledge noise contained in the neighbors. Finally, this paper adopts a multi-task strategy to combine the recommendation task and contrastive learning to strengthen the supervisory signals.

In summary, the contributions of this paper are as follows:

- This paper employs subtle noise following a normal distribution to construct contrasting views, reducing the time overhead of view creation. Through contrastive learning, it further diminishes knowledge noise contained within neighboring information and enhances supervision.
- A knowledge-aware attention mechanism is devised by this study, effectively capturing high-order semantic information in the knowledge graph, thereby mitigating knowledge noise.
- Extensive experiments on three real-world datasets are conducted. The experimental results demonstrate that our model outperforms the baseline models in click-through rate (CTR) and Top-K recommendations, proving the stability of our model.

2 Relation Work

This work involves two different fields: recommendation methods based on knowledge graphs and graph contrastive learning.

2.1 Recommendation Method Based on Knowledge Graph

Knowledge graph-based recommendation approaches predominantly employ the information propagation and aggregation mechanisms of Graph Neural Networks (GNN). By recursively encoding the information of higher-order entities into user and item representations, they emulate the dissemination of user interests. For instance, methodologies such as RippleNet [17], AKUPM [18], and CIEPA [19] leverage the links (relations) within the knowledge graph to propagate users' historical clicked items, thereby exploring users' latent interests. This not only enriches user representations but also enhances recommendation performance. Both KGCN [20] and KGNN-LS [21] utilize Graph Convolutional Networks (GCN) within the knowledge graph to refine item representations, thereby reducing the noise from extraneous knowledge. Such techniques focus solely on enhancing the latent features of users or items through the knowledge graph, without concurrently bolstering both user and item representations. Consequently, Ying et al. [8] introduced the KGAT model, which, by constructing a collaborative knowledge graph, employs both GCN and TransR to capture structural information in the knowledge graph, enhancing both user and item representations. Moreover, Wang et al. [16] presented the CKAN model that balances both the collaborative information of users-items and associations in the knowledge graph. This minimizes the noise introduced by integrating user nodes into the collaborative knowledge graph and underscores the significance of collaborative signals in user and item representations. Building on CKAN, Xu et al. [22] incorporated attention aggregators to discern the significance among different neighbors, enriching the embeddings of users and items. Additionally, Qian et al. [23] proposed the RKAC model to diminish irrelevant knowledge, filtering out redundant knowledge from item attributes.

In summation, extant knowledge graph-based recommendation methodologies largely compute the weights between head entities and relations among neighbors, signifying the importance of tail entities. However, they do not treat the knowledge (triplets) holistically, which could result in a potential loss of semantic information.

2.2 Contrastive Learning on Graph

Contrastive learning (CL) [24,25] has garnered widespread attention in the fields of Computer Vision (CV) and Natural Language Processing (NLP). It effectively enhances embedding quality by minimizing mutual information between positive pairs. As a quintessential self-supervised learning technique, contrastive learning obviates the need for manual annotations and can unearth intrinsic data features to augment downstream tasks. Merging the strengths of Graph Neural Networks (GNN) and CL, researchers have devised numerous graph contrastive learning methods for recommendations, effectively alleviating data sparsity issues and amplifying supervisory signals to better model user interests. For instance, SGL [26] employs node dropping, edge dropping, and random walks to produce multiview representations of a node, maximizing consistency between disparate views of the same node. SimGCL [27], through empirical investigations, discerned the minimal contribution of data augmentation to SGL, proposing the introduction of noise into each embedding layer to generate positive instances, thereby circumventing the manual construction of contrastive instances. XSimGCL [28], building upon SimGCL, discards ineffectual graph augmentations and employs a straightforward noise-based embedding enhancement to create contrasting views.

In summary, existing graph contrastive learning for knowledge-graph-based recommendations predominantly adopts view construction to create positive pairs, potentially leading to the omission of crucial structural information, and adversely impacting the model. Moreover, view construction is time-consuming. Consequently, this paper introduces subtle noise adhering to feature uniformity into the original neighbor embeddings to construct neighbor views, reducing the loss of structural information and time spent on view construction. Simultaneously, by harnessing the self-supervised information from contrastive learning and the supervised information from the recommendation task, this study collectively strengthens the representations of both users and items.

3 Primary Formulation

This paper begins by introducing some relevant symbols and then defines the KG recommendation problem.

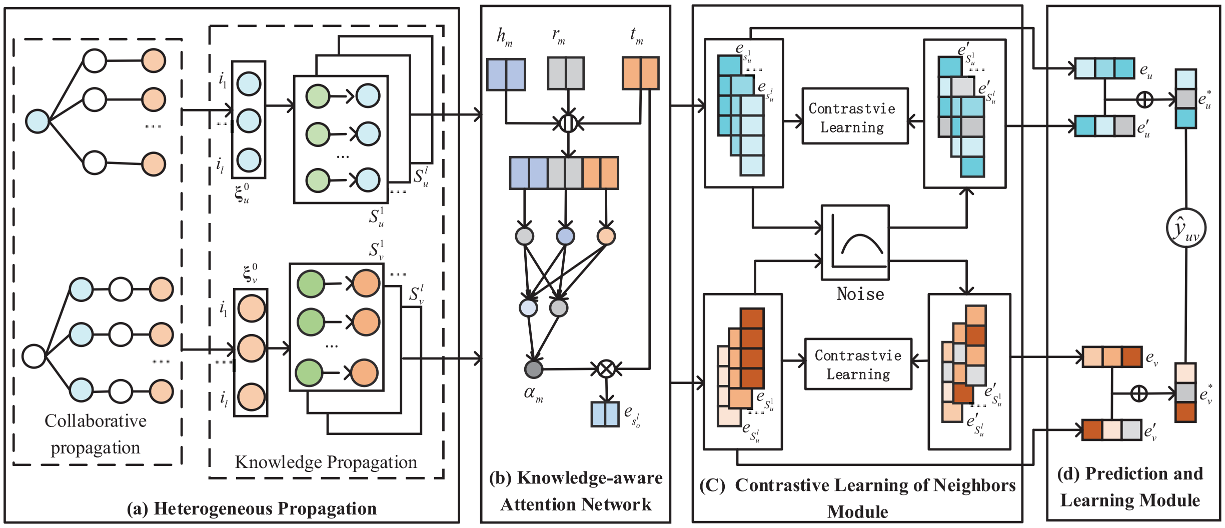
User-Item Graph: In the recommendation context, this paper assumes there are M users and N items. The set of users and items are represented as $\boldsymbol{U} = \{u_1, u_2, \ldots, u_M\}$ and $\boldsymbol{V} = \{v_1, v_2, \ldots, v_N\}$ respectively. If a user interacts with an item, then $y_{uv} = 1$. Otherwise, $y_{uv} = 0$ indicates no interaction.

Knowledge Graph: KG is a heterogeneous network comprised of entities and relationships, which can serve as auxiliary information to enrich the features of users and items in the recommendation system. The formal representation of a knowledge graph is $\boldsymbol{G} = \{(h, r, t) \,|h, t \in \varepsilon, r \in \boldsymbol{R}\}$ where each triplet (h, r, t) denotes a relation r between head entity h and tail entity t, ε is the set of entities, and R is the set of relationships. For example, the triplet ("Hurricane", director, Jizhou Xu) describes the fact that Jizhou Xu directed the TV drama "Hurricane". In many recommendation scenarios, an item $v \epsilon \boldsymbol{V}$ corresponds to an entity $e \in \varepsilon$. Thus, this paper establishes an item-entity alignment $\boldsymbol{A} = \{(v, e) \,| v \in \boldsymbol{V}, e \in \varepsilon\}$, where (v, e) indicates the alignment of item v with entity e in KG. Through the alignment between items and KG entities, KG provides supplementary information for the interaction data between users and items.

Problem Formulation: Given the user-item interaction data and KG $\boldsymbol{G}$, the task of knowledge-aware recommendation is to predict the probability of a user interacting with items he/she has not interacted with before. Specifically, our objective is to learn a prediction function $\hat{y}_{uv} = \boldsymbol{F}(u, v \mid \boldsymbol{\Theta}, \boldsymbol{G})$, where $\hat{y}_{uv}$ represents the predicted probability, and Θ denotes the model parameters of function $\boldsymbol{F}$.

4 Methodology

Our proposed RMCEN aims to reduce noise in KG and enhance the supervisory signal through contrastive learning, thereby improving the representation learning of users and items. Fig. 2 illustrates the workflow of RMCEN, which mainly consists of the following components:



Collaborative propagation
Knowledge Propagation
(a) Heterogeneous Propagation
h_m
r_m
t_m
α_m
(b) Knowledge-aware Attention Network
Contrastvie Learning
Noise
Contrastvie Learning
(C) Contrastive Learning of Neighbors Module
e_u
e'_u
e^*_u
$\hat{y}_{uv}$
e_v
e'_v
e^*_v
(d) Prediction and Learning Module

Figure 2: Illustration of our proposed RMCEN framework

Heterogeneous Propagation: By integrating the user-item graph with KG, this paper demonstrates the connection of user-item interaction information with KG, enabling the acquisition of different levels of neighbor information sets. These sets are generated through the propagation of users and items within KG.

Knowledge-Aware Attention Network: This is used to calculate the weights of knowledge (triplets) in each layer of neighbors, achieving a fine-grained encoding of neighbor information.

Contrastive Learning of Neighbors Module: Purposeful introduction of noise obeying a uniform distribution is performed on the set of neighbor embeddings to generate an augmented set of neighbor embeddings, while contrast learning is employed to learn common features between views to induce consistency of mutual information between views, thus reducing the amount of noise in the neighbor embeddings.

Prediction and Learning Module: This module aggregates the embeddings of each neighbor to get the final representations of users and items. Combined with recommendation tasks and self-supervised learning tasks, it predicts user preferences for items.

4.1 Heterogeneous Propagation

Heterogeneous propagation consists of two parts: collaborative propagation and knowledge propagation. Collaborative propagation constructs the initial entity sets for users and items based on user-item interactions, encoding collaborative signals. On the other hand, knowledge propagation builds upon these initial entity sets, delving into the connections between users and items within the knowledge graph, and propagating neighbor knowledge associations. This method recursively expands the representations of users and items, as shown in Fig. 2a.

4.1.1 Collaborative Propagation

Collaborative propagation extracts key collaborative signals from user-item interaction data and explicitly encodes these signals into representations for users and items. Specifically, user-item interaction data can reflect a user's partial preferences and a user's representation can be manifested through related items. That is, the initial entity set for user u, denoted as $\boldsymbol{\xi}_u^0$, is defined as follows:

$$\boldsymbol{\xi}_u^0 = \{e\,|(v,e) \in \boldsymbol{A}\ and\ v \in \{v\,|y_{uv} = 1\}\} \tag{1}$$

where $\boldsymbol{A}$ represents the item-entity alignment set, and $y_{uv} = 1$ indicates the data where the user interacts with the item.

At the same time, considering the scenario where one item can be interacted with by multiple users, we adopt an "item-user-item" propagation strategy to include user-item interaction information in the initial entity set of the item, enriching the item's representation. The propagation process of "item-user-item" is shown in Fig. 3. Specifically, item i_1 interacts with users u_1 and u_2. User u_1 participates in the interactions with both items i_1 and i_2. Therefore, the collaborative propagation set for item i_1 is $\{i_1, i_2, i_4\}$, meaning the item builds potential item-item relationships through the user as a bridge to obtain the item's initial entity set.

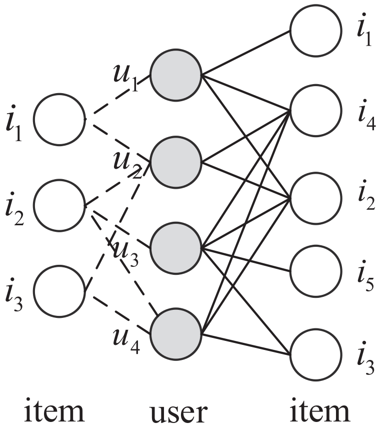


Figure 3: Item-user-item communication process

The initial entity set for item v, denoted as $\boldsymbol{\xi}_v^0$, is defined as follows:

$$\boldsymbol{\xi}_v^0 = \left\{e\,\middle|(v,e) \in \boldsymbol{A}\ and\ v \in \left\{v\,\middle|y_{uv_u} = 1\right\}\right\} \tag{2}$$

where $y_{uv} = 1$ indicates the association between different items of the same user. The collaborative signal of the user-item is explicitly included in the item-item view.

4.1.2 Knowledge Propagation

Knowledge propagation aims to obtain multi-hop neighbor sets of users and items from KG, simulating the latent interests of users and items as they propagate through KG. Specifically, This paper utilizes the initial entity sets of users and items ($\boldsymbol{\xi}_u^0$ and $\boldsymbol{\xi}_v^0$) as seeds. It propagates multiple hops along the relationships in KG to obtain associated entities, subsequently constructing multi-hop entity sets for user u and item v. The entity set $\boldsymbol{\xi}_o^l$ generated from the l-th hop propagation of users and items in KG is defined as:

$$\boldsymbol{\xi}_o^l = \left\{t \,\middle|\, (h, r, t) \in G \textit{ and } h \in \boldsymbol{\xi}_o^{l-1}\right\}, l = 1, 2, \ldots, L \tag{3}$$

where o is a unified placeholder for either user u or item v. $\boldsymbol{G}$ represents the knowledge graph, and (h, r, t) represents a triplet in the knowledge graph. $\boldsymbol{\xi}_o^l$ expresses the retrieval of associated tail entities t in the KG with $\boldsymbol{\xi}_o^{l-1}$ as the head entity for the l-th hop.

To capture structural and semantic information from KG, we take the associated entities $\boldsymbol{\xi}_o^l$ generated from knowledge propagation as the head entity set and retrieves each triplet as the neighbor set for users and items. For the neighbor set $\boldsymbol{S}_o^l$ of the l-th hop for users or items, when the head entity set is defined as $\boldsymbol{\xi}_o^l$, we search for corresponding relationships and tail entities in the KG's triplets to form user and item neighbor sets. The l-th hop neighbor set $\boldsymbol{S}_o^l$ is defined as follows:

$$\boldsymbol{S}_o^l = \left\{(h, r, t) \,\middle|\, (h, r, t) \in \boldsymbol{G} \textit{ and } h \in \boldsymbol{\xi}_o^{l-1}\right\}, l = 1, 2, \ldots, L \tag{4}$$

where $\boldsymbol{S}_o^l$ consists of the triplet sets formed with entity set $\boldsymbol{\xi}_o^{l-1}$ as the head entity.

Knowledge propagation establishes a connection between user-item interaction data and the knowledge graph. The neighbor sets $\left\{\boldsymbol{S}_o^1, \boldsymbol{S}_o^2, \cdots, \boldsymbol{S}_o^L\right\}$ serve to mine the structural and semantic information from KG, thereby enhancing the feature representation of both users and items.

4.2 Knowledge-Aware Attention Network

To explore the importance of neighbor sets $\left\{\boldsymbol{S}_o^1, \boldsymbol{S}_o^2, \cdots, \boldsymbol{S}_o^L\right\}$ in the representation of users and items, we fully consider the varying preferences of the tail entities under different relational links for both users and items. A knowledge-aware attention network is proposed, designed to calculates the attention weight of each triplet in the neighbor set, revealing the variability in the expression meanings of different triplets, effectively encoding the knowledge in each neighbor, as shown in Fig. 2b.

For the m-th triplet (h_m, r_m, t_m) in $\boldsymbol{S}_o^l$, the knowledge-aware attention weight $\rho\,(h_m, r_m, t_m)$ is computed as:

$$c_0 = LeakReLU\left(W_0\left(e_h^m \,\|e_r^m\, \|e_t^m\right) + d_0\right) \tag{5}$$

$$\rho\,(h_m, r_m, t_m) = \sigma\,(W_2 LeakRuLU\,(W_1 c_0 + d_1)) \tag{6}$$

where $\rho(h_m, r_m, t_m)$ is used to gauge the significance of each triplet in the neighbors, thereby capturing knowledge associations more effectively. *LeakReLU* is a non-linear activation function that prevents gradient vanishing. $\boldsymbol{\sigma}$ represents the *sigmoid* activation function, and $\|$ is the concatenation operation. Additionally, a *softmax* function is used to normalize the coefficients a_m of the entire triplet, as follows:

$$\rho\,(h_m, r_m, t_m) = \frac{exp\,(\rho\,(h_m, r_m, t_m))}{\sum_{((h_{m'}, r_{m'}, t_{m'}) \in S_o^l} exp\,(\rho(h_{m'}, r_{m'}, t_{m'}))} \tag{7}$$

where $\boldsymbol{S}_o^l$ designates the l-th layer neighbors of the user or item. Finally, the embedding of the neighbor $\boldsymbol{S}_o^l$ can be represented as:

$$e_{S_o^l} = \sum\nolimits_{m=1}^{S_o^l} \rho\left(h_m, r_m, t_m\right) t_m \tag{8}$$

where $e_{S_o^l}$ is the embedding of the neighbor $\boldsymbol{S}_o^l$.

The neighbor sets $\left\{\boldsymbol{S}_o^1, \boldsymbol{S}_o^2, \cdots, \boldsymbol{S}_o^L\right\}$, after the knowledge-aware attention calculation, acquire embeddings that can be represented as $\left\{e_{S_o^1}, e_{S_o^2}, \cdots, e_{S_o^L}\right\}$. Because the knowledge-aware attention network only focuses on fine-grained knowledge (triplets) to reduce knowledge noise and neglects the noise contained in the structure of the neighbors, we introduce contrastive learning to address the imbalance in the structure of the neighbors, thereby enhancing their embedding.

4.3 Contrastive Learning of Neighbor Information

Contrastive learning is a self-supervised learning task that does not rely on labeled data from supervised learning but focuses on the intrinsic features of the data. Based on two key factors identified in the application of contrastive learning in the CV (Computer Vision) and NLP (Natural Language Processing) domains: feature alignment and feature uniformity constraints. Therefore, we introduce minor noise that conforms to feature uniformity into the original neighbor embeddings to construct neighbor views, as shown in Fig. 2c. At the same time, a contrastive loss function is used between neighbor views to adjust the neighbor structure, thereby reducing noise in the item neighbors.

4.3.1 Neighbor View Construction

For the neighbor embedding set of the project $\left\{e_{S_v^1}, e_{S_v^2}, \cdots, e_{S_v^L}\right\}$, we introduce noise following a uniform distribution to simulate the uniformity of the features. For the i-th hop neighbor embedding of item $e_{S_v^i}$, its enhanced neighbor embedding $e'_{S_v^i}$ is as follows:

$$e'_{S_v^i} = e_{S_v^i} + \Delta_{noise} \tag{9}$$

where Δ_{noise} is the added noise vector, and $\|\Delta_{noise}\| = \epsilon$ is a small constant ($\epsilon < 0.3$). To restrict Δ_{noise} to points on a hyper-sphere with a radius equivalent to ϵ, the equation is:

$$\Delta_{noise} = I \odot sign\left(e_{S_v^i}\right), I \in R^d \sim U\left(0, 1\right) \tag{10}$$

The $e'_{S_v^i}$ constructed through noise Δ_{noise} retains most of the original features and introduces some variations. Additionally, the same approach is used for the neighbor embedding set of the users.

Compared to using dropout in neighbor embeddings which results in a loss of some features, we introduce minute noise to construct the neighbor view. This enhanced neighbor view follows feature uniformity and does not lose any features. Furthermore, our constructed enhanced neighbor view, when compared to neighbor views built through structural perturbation, incurs no additional time overhead in view construction.

4.3.2 Contrastive Loss Function

To maximize the mutual information between neighbor views, we employ InfoNCE [23] as the contrastive loss function, aiming to reduce the knowledge noise in neighbor information and enhance the representations of users and items. Specifically, for the original neighbor embedding set of the item $\left\{e_{S_v^1}, \ldots e_{S_v^k}, \ldots, e_{S_v^L}\right\}$, through neighbor view construction, it can be represented as $\{e'_{S_v^1}, \ldots, e'_{S_v^k}, \ldots, e'_{S_v^L}\}$. Embeddings of neighbors of the same order are considered positive pairs (i.e., $e_{S_v^k}$ and $e'_{S_v^k}$), while embeddings of neighbors of different orders are viewed as negative pairs (i.e., $e_{S_v^k}$ and

$e'_{s_v^{j\neq k}}$). Through contrasting between positive and negative pairs, a self-supervisory signal is generated to mitigate the issue of weak supervisory signals. We adopt the InfoNCE [29] function as the contrastive loss function for the contrastive learning task:

$$L_{InfoNCE} = -\sum\nolimits_{k\in L} log \frac{\exp\left(e'_{s_v^k} \cdot e_{s_v^k}/\tau\right)}{\sum_{k\in L} \exp\left(e'_{s_v^k} \cdot e_{s_v^k}/\tau\right)} \tag{11}$$

where $s\left(,\right)$ denotes the cosine similarity, and τ is the temperature parameter, regulating the uniform distribution of samples. Introducing additional supervisory information through InfoNCE encourages the item neighbor embeddings to spread uniformly across the latent space, facilitating better distinction during predictions. The same processing method is applied to users.

4.4 Prediction and Learning

To delve into the personalized preferences of users, we use the dot product to compute the scores of users for items, as shown in Fig. 2d. Furthermore, we employ a multi-task learning strategy to jointly address recommendation tasks and self-supervised learning tasks, thereby optimizing the model.

4.4.1 Model Prediction

Considering that the initial entity sets of users and items ($\boldsymbol{\xi}_u^0$ and $\boldsymbol{\xi}_v^0$) contain collaborative information, their collaborative embeddings are represented as $e_{s_u^0}$ and $e_{s_v^0}$. To retain all the information maximally, we concatenate the neighbor embedding set $\left\{e_{s_o^1}, e_{s_o^2}, \ldots, e_{s_o^L}\right\}$ with the collaborative embeddings to obtain the final embeddings of users and items. The equation is as follows:

$$e_o = e_{s_o^0}||e_{s_o^1}||\ldots||e_{s_o^L} \tag{12}$$

where o is a unified placeholder that represents the features of users u and items v. Furthermore, considering that the enhanced neighbor views of users and items have weak noise added, we directly use the enhanced neighbor embedding set $\{e'_{s_o^1}, e'_{s_o^2}, \ldots, e'_{s_o^L}\}$ for the recommendation task. The final representations of users and items are denoted as e_u^* and e_v^*, respectively. The equation is:

$$e_u^* = e'_u + e_u \tag{13}$$

$$e'_u = e'_{s_u^1}||e'_{s_u^2}||\ldots||e'_{s_u^L} \tag{14}$$

$$e_v^* = e'_v + e_v \tag{15}$$

$$e'_v = e'_{s_v^1}||e'_{s_v^2}||\ldots||e'_{s_v^L} \tag{16}$$

Based on the final user representation e_u^* and item representation e_v^*, the user preference function $\hat{y}\left(u, v\right)$ is:

$$\hat{y}\left(u, v\right) = e_v^{*T} e_v^* \tag{17}$$

4.4.2 Multi-Task Learning

We frame our primary objective as a supervised learning-based recommendation task while incorporating contrastive learning based on self-supervised learning as an auxiliary task. By adopting a joint learning approach, we introduce self-supervised signals into the recommendation model to augment the supervisory signal.

For recommendation tasks, we employ the Bayesian Personalized Ranking (BPR) loss function, as follows:

$$L_{BPR} = \sum\nolimits_{(u,i,j) \in O} -ln\boldsymbol{\sigma} \left(\hat{y}_{ui} - \hat{y}_{uj} \right) \tag{18}$$

where $O = \{(u, i, j) | (u, i) \in o^+,\ (u, i) \in o^-\}$ is the trainable dataset, user-item interactions are denoted as o^+, non-interactions between the user and the item are denoted as o^-, and the data ratio of o^+ to o^- is set to 1:1. $\boldsymbol{\sigma}$ represents the sigmoid function.

For the recommendation task, we employ joint learning, combining both supervised and self-supervised signals to collectively optimize the model. The final loss function for this model is formulated as follows:

$$L = L_{BPR} + \lambda_1 L_{InfoNCE} + \lambda_2 \left\| \Theta \right\|_2^2 \tag{19}$$

where $\boldsymbol{\Theta}$ denotes the set of model parameters, and $\left\| \Theta \right\|_2^2$ is the L2 norm with λ_2 as its parameter.

4.4.3 Complexity Analysis

The training time cost for the RMCEN model predominantly stems from heterogeneous propagation and contrastive learning. Specifically, the computational complexity of heterogeneous propagation is $(|G^K|d)$, where L, $|G^K|$, and d represent the number of layers, the number of triplets in the KG, and the embedding size, respectively. In contrast, the computational complexity for contrastive learning is $(|B_U + B_I|d)$, where B_U and B_I respectively denote the number of user and item triplets in a batch. Consequently, the overall computational complexity of the RMCEN model is $(|G^K|d + |B_U + B_I|d)$.

Based on the aforementioned analysis and under equivalent experimental settings, the RMCEN model exhibits a complexity comparable to most knowledge graph-based recommendation methods in the comparison, including KGCN, KGAT, COAT, and CKAN. Meanwhile, its complexity is significantly lower than that of the KGIN model in the comparative set.

5 Experiments

We evaluated the proposed RMCEN on three real-world public datasets, namely MovieLen-1M (movie recommendation scenario), Book-Crossing (book recommendation scenario), and Last.FM (music recommendation scenario), to answer the following questions:

RQ1: How does the RMCEN model perform in comparison to other recommendation methods in the recommendation task?

RQ2: Are the various components of the RMCEN model effective?

RQ3: How do different hyperparameter settings affect the results?

5.1 Datasets Description

To assess the effectiveness of our method, we conducted experiments on the Movielen-1M, Book-Crossing, and Last.FM datasets. Detailed information about these datasets is shown in Table 1.

- MovieLens-1M is one of the widely used datasets for movie recommendations. It includes ratings of movies, metadata about the movies (such as genres and release years), and descriptive data about users (including age, gender, and occupation).
- MovieLens-20M is a benchmark dataset widely employed in the realm of film recommendations. It encompasses approximately 20 million explicit user ratings, spanning from 1 to 5.

- Last.FM is a popular recommendation dataset extracted from an online music provider. It contains records of tracks that many users have listened to, along with metadata about the listeners and the music.
- Book-Crossing is a book rating dataset that contains ratings of books as well as descriptions of the readers and the books.

Table 1: Data set statistics

	MovieLens-1M	MovieLens-20M	Book-Crossing	Last.FM
user	6,036	138159	17,860	1,872
item	2,445	16954	14,967	3,846
interaction	753,772	13501622	139,746	42,346
sparsity	5.11%	0.576%	0.05%	0.59%
entity	182,011	102569	77,903	9,366
relation	12	32	25	60
triplet	1,241,996	499474	1,515,00	15,518

5.2 Parameter Settings

For each dataset, we divided the data into training, validation, and test sets in a 6:2:2 ratio. The Adam optimizer was employed for model optimization, with the batch size set to 2048. Xavier initialization was used for initializing the model parameters. Other important hyperparameter settings are outlined in Table 2.

Table 2: Parameter settings in the data set

Parameter settings	MovieLens-1M	MovieLens-20M	Book-Crossing	Last.FM
Dimensions (D)	64	64	64	64
Number of layers (H)	1	1	2	2
Number of neighbors	64	64	64	64
Control parameters λ_1	10−7	10−7	10−5	10−4
Learning rate λ_2	0.004	0.004	0.004	0.004
Temperature parameter (τ)	0.15	0.15	0.15	0.15
Noise parameters (ϵ)	0.1	0.1	0.1	0.1

5.3 Baselines

- BPRMF [30]: This is a Collaborative Filtering (CF) based model that employs matrix factorization and Bayesian Personalized Ranking to predict user preferences.
- PER [31]: This is a path-based recommendation method. It considers KG as a heterogeneous network and extracts meta-paths and meta-graphs containing different types of relationships to embed users and items.
- CKE [32]: This is a classical embedding-based recommendation model that uses TransR to learn knowledge in KG, enriching the representation of all items.

- RippleNet [17]: This method simulates the ripple propagation mechanism to propagate user preferences along the relationships in KG, enhancing user modeling.
- KGCN [20]: This method integrates Knowledge Graphs and Graph Convolutional Networks (GCNs). By traversing each node's neighbors in KG, it computes the GCN-based neighbor information to learn both the structural and semantic information in KG, capturing users' latent interests.
- KGAT [14]: This method combines user-item graphs with knowledge graphs as a collaborative knowledge graph and employs attention-based neighborhood aggregation mechanisms to generate representations for users and items.
- COAT [33]: This method uses both Graph Convolutional Networks and Knowledge Graph Attention Networks to model user-item graphs and KG separately, encoding collaborative signals and capturing fine-grained semantic information in KG.
- LKGR [34]: This method employs different information propagation strategies in hyperbolic space to encode both the interaction information between users and items and the heterogeneous information in KG.
- KGIN [15]: This method explores the intentions behind user-item interactions and model relationships in KG within the context of user intentions to achieve improved model capabilities and explainability.
- CG-KGR [35]: This method encodes collaborative information between users and items through a co-guidance mechanism and uses it to guide the encoding of knowledge in KG.
- KGIC [36]: This method introduces multi-level interactive contrastive learning, mining features from both local and non-local graphs of users and items, to alleviate data sparsity and knowledge redundancy.

5.4 Performance Comparison (RQ1)

5.4.1 Performance Comparison in CTR Prediction Task

Table 3 presents the AUC and F1 scores of RMCEN and all baseline models in the task of Click-Through Rate (CTR) prediction. Based on the performance analysis, we make the following observations:

- Our proposed RMCEN model achieves the best performance across all three datasets. Specifically, on the MovieLens-1M, Book-Crossing, and Last.FM datasets, RMCEN outperforms the best baseline (highlighted with an underline) in terms of AUC by 0.28%, 1.31%, and 1.33%, respectively. This validates the effectiveness of RMCEN. We attribute these improvements to the neighbor contrastive learning implemented in RMCEN.
- The performance improvements in RMCEN are more noticeable on the Book-Crossing and Last.FM datasets compared to the MovieLens-1M dataset. This is because Book-Crossing and Last.FM are sparser than MovieLens-1M, suggesting that the introduction of Knowledge Graphs (KGs) enriches the latent representations of users and items in sparse datasets with additional features.
- In two distinct datasets, MovieLens-1M and MovieLens-20M, our proposed RMCEN exhibits a noticeable performance enhancement on the MovieLens-1M dataset compared to the MovieLens-20M dataset. This discrepancy in performance arises due to the dense user-item interactions present in the MovieLens-20M dataset, where the introduction of a knowledge graph does not significantly improve the recommendation performance.

- RippleNet focuses only on propagating user preferences in KG and lacks rich item information. Similarly, KGCN and KGCN-LS aggregate neighbor information to enrich item embeddings but neglect the user representation. Our RMCEN model takes into account heterogeneous propagation to obtain neighbors for both users and items in KG, thereby enriching both user and item representations.
- Methods like KGAT, CKAN, COAT, KGIN, CG-KGR, etc., aim to enrich both user and item embeddings through KGs, but their focus is on the design of propagation and aggregation strategies, without considering the large amount of noise present in KGs. Our RMCEN uses contrastive learning to minimize the noise contained in the neighbors of users and items, thus reducing the noise in the aggregated data. Compared to KGIC, which builds multi-level interaction views for contrastive learning at the cost of high time complexity, our proposed RMCEN model adds only weak noise to construct enhanced neighbor embeddings, thus reducing the time complexity involved in building neighbor views for users and items.

Table 3: Parameter settings in the data set

Method	MovieLens-1M		MovieLens-20M		Book-Crossing		Last.FM	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
BPRMF	0.8920	0.7921	0.9574	0.9128	0.6583	0.6117	0.7563	0.7010
PER	0.7124	0.6670	0.8325	0.8025	0.6048	0.5726	0.6414	0.6033
CKE	0.9065	0.8024	0.9236	0.8739	0.6749	0.6234	0.7471	0.6740
RippleNet	0.9102	0.8463	0.9673	0.9266	0.7263	0.6472	0.7938	0.7025
KGCN	0.9001	0.8261	0.9767	0.9302	0.6922	0.6336	0.8031	0.7107
KGNN-LS	0.9140	0.8410	0.9785	0.9322	0.6890	0.6334	0.8042	0.7283
KGAT	0.9102	0.8404	0.9813	0.9249	0.7194	0.6544	0.8054	0.7224
CKAN	0.9170	0.8494	0.9717	0.9230	0.7453	0.6667	0.8463	0.7715
COAT	0.9122	0.8367	0.9784	0.9313	0.7407	0.6611	0.8215	0.7499
LKGR	0.9178	0.8419	0.9784	0.9332	0.6793	0.6296	0.7963	0.7219
KGIN	0.9190	0.8441	0.9820	0.9335	0.7273	0.6614	0.8463	0.7602
CG-KGR	0.9110	0.8359	0.9783	0.9306	0.7498	0.6689	0.8336	0.7433
KGIC	0.9252	0.8510	0.9763	0.9312	0.7652	0.6754	0.8466	0.7753
RMCEN (our)	**0.9280**	**0.8650**	**0.9831**	**0.9340**	**0.7743**	**0.6885**	**0.8599**	**0.7817**

5.4.2 Performance Comparison for Top-K Recommendation Task

For the Top-K recommendation task, Fig. 4 shows the Recall@K on the MovieLen-1M, Book-Crossing, and Last.FM datasets, where K is set to [5,10,20,50,100].

From Fig. 4, it is evident that our method performs best overall across the three datasets. Compared with the best models on Last.FM, Book-Crossing and MovieLens-1M, our method shows significant improvement. In Figs. 2a and 2c, the Recall@K values have significantly increased, indicating that the introduction of contrastive learning can differentiate between hard-to-distinguish samples among item neighbors, thereby reducing knowledge noise. In Fig. 2b, our method is close to CKAN, which may be because the corresponding knowledge graph in MovieLens-1M contains less noise, making the introduction of contrastive learning less effective in improving its recommendation performance.

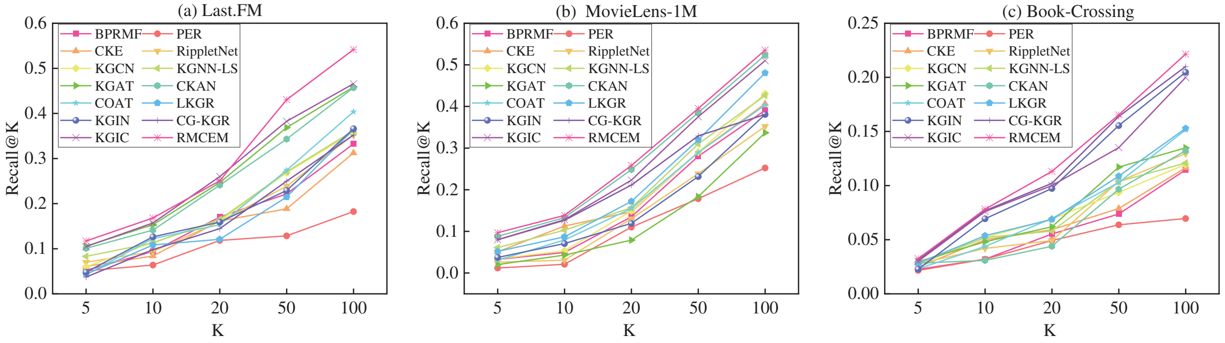


Figure 4: Recall@K in three data sets

5.5 Ablation Study of RMCEN (RQ2)

To investigate the impact of heterogeneous propagation, knowledge-aware attention mechanism, and contrastive learning on recommendation performance, we conduct the following ablation experiments:

(1) RMCEN $_{\text{w/o CL}}$: In this variant, the contrastive learning module is removed, retaining the original neighbor information of users and items.

(2) RMCEN $_{\text{w/o att}}$: This variant removes the knowledge-aware attention mechanism and replaces it in Eq. (6).

(3) RMCEN $_{\text{w/o CL-I}}$: This variant removes the item neighbor data augmentation module, retaining only the contrastive learning for users.

(4) RMCEN $_{\text{w/o CL-U}}$: This variant removes the user neighbor data augmentation module, retaining only the contrastive learning for items.

(5) RMCEN $_{\text{w/o drop}}$: This variant removes the neighbor view construction module and uses dropout to randomly remove some neighbor features, constructing neighbors.

Table 4 shows the results of the four variants and RMCEN, and we make the following observations:

- Removing the contrastive learning module (RMCEN w/o CL) in all three datasets results in a decline in model performance. This indicates that contrastive learning plays a significant role in our method. The introduction of contrastive learning can adaptively adjust the structure of item neighbors, reducing the noise in the item neighbors and improving recommendation performance.
- Disabling the knowledge-aware attention mechanism (RMCEN w/o att) in all three datasets leads to a significant decline in model performance. This indicates that our proposed knowledge-aware attention mechanism is effective at capturing important knowledge among neighbors and minimizing the interference of irrelevant triples.
- Removing the item neighbor data augmentation module (RMCEN w/o CL-I) and deleting the user neighbor data augmentation module (RMCEN w/o CL-U) on all three datasets results in

a significant decline in recommendation performance. This suggests that applying contrastive learning to both user and item neighbors is necessary.

- In all three datasets, using dropout technology to randomly remove some neighbor features leads to a significant decline in recommendation performance. This is because the random removal of some neighbor features might result in the loss of some crucial knowledge. This validates that our introduction of weak noise following a normal distribution to enhance neighbor embeddings can effectively mitigate the loss of knowledge.

In summary, all the modules or techniques we proposed are necessary; removing any part will lead to a decrease in recommendation performance.

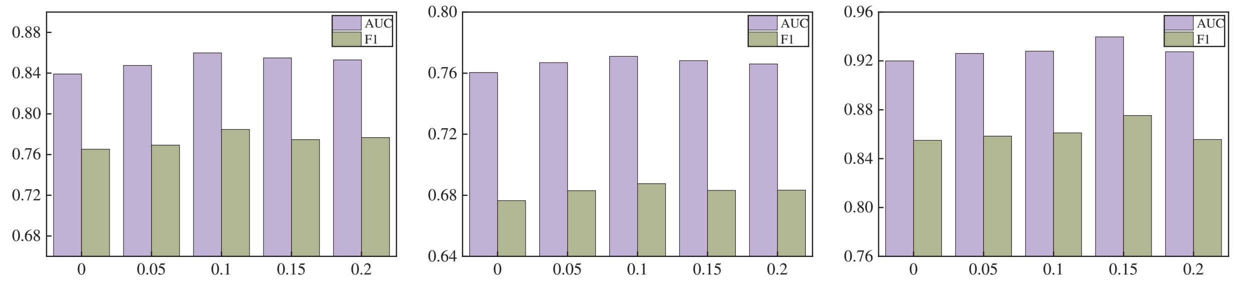
Table 4: Parameter settings in the data set

	MovieLens-1M		Book-Crossing		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
RMCEN $_{\text{w/o CL}}$	0.9250	0.8544	0.7481	0.6665	0.8447	0.7692
RMCEN $_{\text{w/o CL-U}}$	0.9269	0.8553	0.7607	0.6792	0.8534	0.7765
RMCEN $_{\text{w/o CL-I}}$	0.9274	0.8563	0.7648	0.6777	0.8549	0.7751
RMCEN $_{\text{w/o att}}$	0.9188	0.8498	0.7422	0.6620	0.8427	0.7622
RMCEN $_{\text{w/o drop}}$	0.9208	0.8510	0.7560	0.6675	0.8386	0.7636
RMCEN	**0.9280**	**0.8650**	**0.7743**	**0.6885**	**0.8599**	**0.7817**

5.6 *Parameter Sensitivity Analysis (RQ3)*

5.6.1 *Size of Noise* ϵ

To explore the optimal range of noise, this paper sets ϵ to [0, 0.05, 0.1, 0.15, 0.2], as shown in Fig. 5.



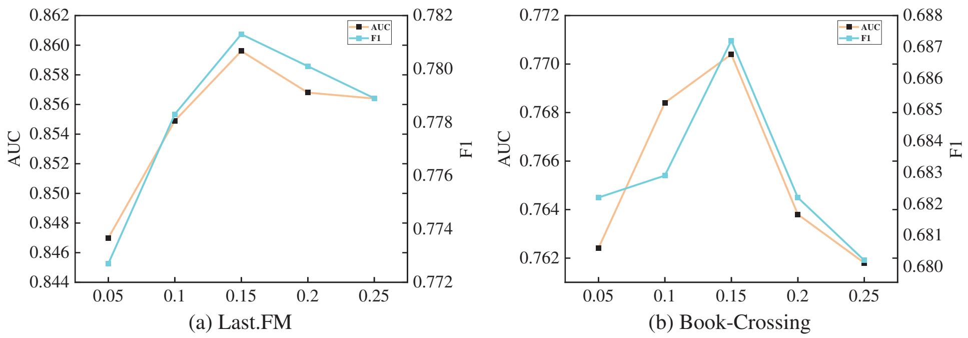
(a) Last.FM (b) Book-Crossing (c) MovieLen-1M

Figure 5: Effect of noise ϵ

In Fig. 5, our model achieves the best performance on the Last.FM, Book-Crossing, and MovieLens-1M datasets when set to [0.1, 0.1, 0.15]. Without adding noise ($\epsilon = 0$), we find that performance drops significantly. When the noise is too large ($\epsilon = 0.2$) or too small ($\epsilon = 0.05$), performance also declines. This validates that adding noise appropriately can make the data distribution more uniform, which is beneficial for improving performance.

5.6.2 *Impact of Temperature τ*

According to existing contrastive learning works [23], the temperature τ defined in Eq. (12) plays a crucial role in contrastive learning. To study the impact of τ, we vary its value within the range {0.05, 0.1, 0.15, 0.2, 0.25}, as shown in Fig. 6.



(a) Last.FM (b) Book-Crossing

Figure 6: Effect of temperature τ

The results shown in Fig. 6 indicate that satisfactory recommendation performance can be obtained when $\tau = 0.15$. When the value of τ is too large, the recommendation performance is not good, which is consistent with the conclusions of previous work [36]. This suggests that τ can regulate the uniformity of the learned neighbor features in the embedding space.

5.6.3 *Depth of the Propagation Layers*

To investigate the impact of the model's propagation depth, we vary it within the range L = {1, 2, 3} where L represents the number of propagation layers. Table 5 compares the performance on the MovieLens-1M, Book-Crossing, and Last.FM datasets.

Table 5: The impact of the number of layers in the three data sets

Layers	MovieLens-1M		Book-Crossing		Last.FM	
	AUC	F1	AUC	F1	AUC	F1
L = 1	**0.9280**	**0.8650**	0.7684	0.6831	0.8492	0.7728
L = 2	0.9272	0.8571	**0.7743**	**0.6885**	**0.8599**	**0.7817**
L = 3	0.9256	0.8532	0.7638	0.6815	0.8561	0.7767

Table 5 shows that when L = 1, 2, 2, RMCEN performs best on the MovieLens-1M, Book-Crossing, and Last.FM datasets. This suggests that aggregating neighbor information in the graph at 1 or 2 layers is an appropriate distance, and stacking more layers further would lead to a decline in recommendation performance. We believe there are two reasons for this result. First, increasing the number of stacked layers will only introduce more noise. Second, the model structure of RMCEN is complex, and an increase in the number of layers may lead to over-smoothing and overfitting.

5.6.4 *Embedding Dimension*

This paper varies D in [8, 16, 32, 64, 128] to analyze the impact of the embedding dimension on the recommendation performance, as shown in Fig. 7.

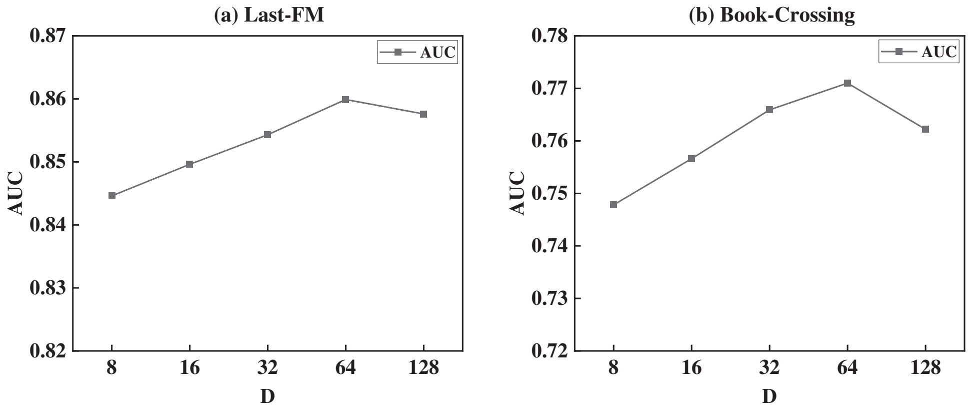


Figure 7: Impact of dimension D on recommendation performance

From Fig. 7, it can be seen that increasing dimension D can improve recommendation performance, as a larger latent vector space can retain more rich information. However, if the dimension d is too large, RMCEN will experience overfitting.

5.7 *Sparse Analysis*

In the original dataset, this paper randomly removed user-item interaction data at different proportions of 10%, 30%, and 50% to simulate the impact of different levels of data sparsity on the model's performance, as shown in Fig. 8.

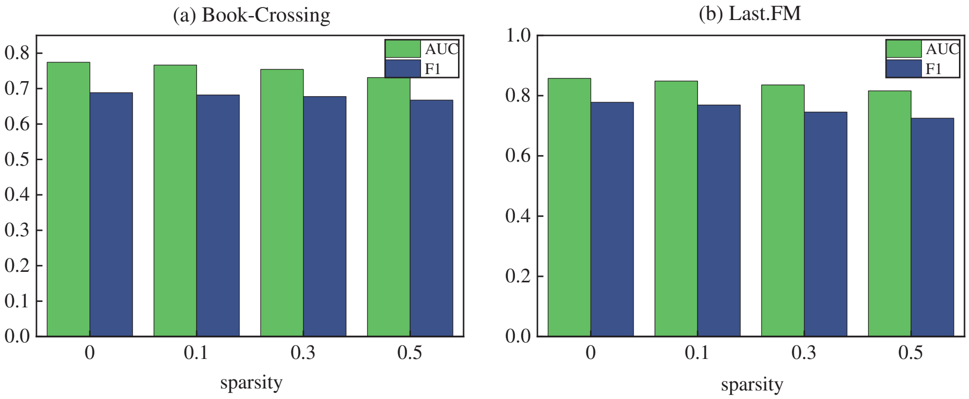


Figure 8: Impact of different levels of sparsity

The experimental results from Fig. 8 indicate that the performance of our proposed RMCEN remains relatively stable under varying degrees of data sparsity.

6 Conclusion

This paper explores the application of contrastive learning in knowledge graph-based recommendation methods and proposes a recommendation method enhanced by neighboring information contrast. In this method, a knowledge-aware attention mechanism is designed to learn the information of triples in neighbors, which helps in reducing knowledge noise and strengthening the representation of users and items. Minor noise, following a normal distribution, is introduced into the neighboring features to create a contrasting view. Through contrastive learning, the method aims to reduce knowledge noise in neighboring features and strengthen the supervisory signal. The effectiveness of this method is validated through extensive experiments on CTR prediction and Top-K recommendation tasks.

Our method only considers the knowledge graph of the project and does not introduce the social network of users. In future work, we plan to integrate KG and social networks for modeling, which can better understand users and items and provide more personalized recommendations. Furthermore, regarding the temperature parameter in contrastive learning, our method uses a manual setting, which may not find the most suitable value. In subsequent research, it is considered to adjust the temperature parameter automatically to find the most suitable value for specific data, which will help improve recommendation performance.

Acknowledgement: None.

Funding Statement: This work was supported by the Natural Science Foundation of Ningxia Province (No. 2023AAC03316), the Ningxia Hui Autonomous Region Education Department Higher Education Key Scientific Research Project (No. NYG2022051), the North Minzu University Graduate Innovation Project (YCX23146).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Hairong Wang, Beijing Zhou; data collection: Beijing Zhou; analysis and interpretation of results: Beijing Zhou, Lisi Zhang; draft manuscript preparation: Hairong Wang, Beijing Zhou, Ma He. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets generated during and/or analyzed during the current study are available in the https://grouplens.org/datasets/movielens/; https://grouplens.org/datasets/book-crossing/; http://millionsongdataset.com/lastfm/ repository.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu *et al.,* "Neural collaborative filtering," in *Proc. of the 26th Int. Conf. on World Wide Web*, Perth, Australia, pp. 173–182, 2017.

[2] I. Chen, H. Zhang, X. He, L. Nie, L. W. Liu *et al.,* "Attentive collaborative filtering: Multimedia recommendation with item—and component-level attention," in *Proc. of the 40th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Tokyo, Japan, pp. 335–344, 2017.

[3] H. Chen, Y. Li, S. Shi, S. Liu, H. Zhu *et al.,* "Graph collaborative reasoning," in *Proc. of the Fifteenth ACM Int. Conf. on Web Search and Data Mining*, Arizona, USA, pp. 75–84, 2022.

[4] D. Zou, W. Wei, X. L. Mao, Z. Wang, M. Qiu *et al.,* "Multi-level cross-view contrastive learning for knowledge-aware recommender system," in *Proc. of the 45th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Madrid, Spain, pp. 1358–1368, 2022.

[5] C. Ma, Y. G. Sun, Z. G. Yang, H. Huang, D. Y. Yang *et al.,* "Content feature extraction-based hybrid recommendation for mobile application services," *Computers, Materials & Continua*, vol. 71, no. 3, pp. 6201–6217, 2022.

[6] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo *et al.,* "SHINE: Signed heterogeneous information network embedding for sentiment link prediction," in *Proc. of the Eleventh ACM Int. Conf. on Web Search and Data Mining*, Los Angeles, California, USA, pp. 592–600, 2018.

[7] J. Wang, Y. Shi, H. Yu, Z. Yan, H. Li *et al.,* "A novel KG-based recommendation model via relation-aware attentional GCN," *Knowledge-Based Systems*, vol. 275, pp. 110702, 2022.

[8] Y. Ying, F. Zhuang, Y. Zhu, D. Wang and H. Zheng, "CAMUS: Attribute-aware counterfactual augmentation for minority users in recommendation," in *Proc. of the ACM Web Conf. 2023*, Austin, TX, USA, pp. 1396–1404, 2023.

[9] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. of the Fourth ACM Conf. on Recommender Systems*, Barcelona, Spain, pp. 135–142, 2010.

[10] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao *et al.,* "Graph neural networks for social recommendation," in *Proc. of the World Wide Web Conf.*, Taipei, Taiwan, pp. 417–426, 2019.

[11] X. Wang, D. Wang, C. Xu, X. He, Y. Cao *et al.,* "Explainable reasoning over knowledge graphs for recommendation," in *Proc. of the AAAI Conf. on Artificial Intelligence*, Honolulu, Hawaii, USA, pp. 5329–5336, 2019.

[12] Y. Qin, C. Gao, S. Wei, Y. Wang, D. Jin *et al.,* "Learning from hierarchical structure of knowledge graph for recommendation," *ACM Transactions on Information Systems*, vol. 42, no. 1, pp. 1–24, 2023.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2017.

[14] X. Wang, X. He, Y. Cao, M. Liu and T. S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. of the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, pp. 950–958, 2019.

[15] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu *et al.,* "Learning intents behind interactions with knowledge graph for recommendation," in *Proc. of the Web Conf. 2021*, Ljubljana, Slovenia, pp. 878–887, 2021.

[16] Z. Wang, G. Lin, H. Tan, Q. Chen and X. Liu, "CKAN: Collaborative knowledge-aware attentive network for recommender systems," in *Proc. of the 43rd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, China, pp. 219–228, 2020.

[17] H. Wang, F. Zhang, J. Wang, M. Zhao and W. Li, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proc. of the 27th ACM Int. Conf. on Information and Knowledge Management*, Torino, Italy, pp. 417–426, 2018.

[18] X. Tang, T. Wang, H. Yang and H. Song, "AKUPM: Attention-enhanced knowledge-aware user preference model for recommendation," in *Proc. of the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, pp. 1891–1899, 2019.

[19] Y. Lin, B. Xu, J. Feng, H. Lin and K. Xu, "Knowledge-enhanced recommendation using item embedding and path attention," *Knowledge-Based Systems*, vol. 233, pp. 107484, 2021.

[20] H. Wang, M. Zhao, X. Xie, W. Li and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proc. of the World Wide Web Conf.*, Anchorage, AK, USA, pp. 3307–3313, 2019.

[21] H. Wang, F. Zhang, M. Zhang, J. Leskovec and M. Zhao, "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," in *Proc. of the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, pp. 968–977, 2019.

[22] Z. Xu, H. Liu, J. Li, Q. Zhang and Y. Tang, "CKGAT: Collaborative knowledge-aware graph attention network for Top-N recommendation," *Applied Sciences*, vol. 12, no. 3, pp. 1669–1675, 2022.

[23] F. Qian, Y. Zhu, H. Chen, J. Chen, S. Zhao *et al.,* "Reduce unrelated knowledge through attribute collaborative signal for knowledge graph recommendation," *Expert Systems with Applications*, vol. 201, pp. 117078, 2022.

[24] X. Wang, N. Liu, H. Han and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning," in *Proc. of KDD*, Virtual Event, Singapore, pp. 1726–1736, 2021.

[25] Z. Lin, C. Tian, Y. Hou and W. X. Zhao, "Improving graph collaborative filtering with neighborhood-enriched contrastive learning," in *Proc. of the ACM Web Conf. 2022*, Lyon, France, pp. 2320–2329, 2022.

[26] J. Wu, X. Wang, F. Feng, X. He and L. Chen, "Self-supervised graph learning for recommendation," in *Proc. of the 44th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Canada, pp. 726–735, 2021.

[27] J. Yu, H. Yin, X. Xia, T. Chen and L. Cui, "Are graph augmentations necessary?: Simple graph contrastive learning for recommendation," in *Proc. of the 45th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Madrid, Spain, pp. 1294–1303, 2022.

[28] J. Yu, X. Xia, T. Chen, L. Cui, N. Q. V. Hung *et al.,* "XSimGCL: Towards extremely simple graph contrastive learning for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, pp. 1–14, 2023.

[29] A. V. D. Oord, Y. Li and O. Vinyals, "Representation learning with contrastive predictive coding," arXiv preprint arXiv:1807.03748, 2018.

[30] Q. Dai, X. M. Wu, L. Fan, Q. Li, H. Liu *et al.,* "Personalized knowledge-aware recommendation with collaborative and attentive graph convolutional networks," *Pattern Recognition*, vol. 128, pp. 108628, 2022.

[31] S. Rendle, C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," arXiv preprint arXiv:1205.2618, 2012.

[32] X. Yu, X. Ren, Y. Sun, Q. Gu and B. Sturt, "Personalized entity recommendation: A heterogeneous information network approach," in *Proc. of the 7th ACM Int. Conf. on Web Search and Data Mining*, New York, NY, USA, pp. 283–292, 2014.

[33] F. Zhang, N. J. Yuan, D. Lian, X. Xie and W. Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, pp. 353–362, 2016.

[34] Y. Chen, M. Yang, Y. Zhang, M. Zhao and Z. Meng, "Modeling scale-free graphs with hyperbolic geometry for knowledge-aware recommendation," in *Proc. of the Fifteenth ACM Int. Conf. on Web Search and Data Mining*, AZ, USA, pp. 94–102, 2022.

[35] Y. Chen, Y. Yang, Y. Wang, J. Bai and X. Song, "Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation," in *Proc. of IEEE 38th Int. Conf. on Data Engineering*, Kuala Lumpur, Malaysia, pp. 299–311, 2022.

[36] D. Zou, W. Wei, Z. Wang, X. L. Mao and F. Zhu, "Improving knowledge-aware recommendation with multi-level interactive contrastive learning," in *Proc. of the 31st ACM Int. Conf. on Information & Knowledge Management*, Atlanta, GA, USA, pp. 2817–2826, 2022.