CLNSO: A Knowledge-Aware Recommendation Algorithm Based on Comparative Learning and Negative Sample Optimization

Jie Yang, Wenhua Cui*, Ye Tao, Tianwei Shi

Abstract—**Although existing methods utilize knowledge graphs and comparative learning to improve recommendation performance, they are still deficient in negative sample selection and information aggregation and are difficult to effectively deal with data sparsity and noise problems. In this paper, we propose a knowledge-aware recommendation algorithm based on comparative learning and negative sample optimization (CLNSO). The model is based on a user-item interaction matrix and knowledge graph and realizes the fusion of multi-level knowledge information by constructing local and non-local graphs. The model adopts an adaptive negative sample selection strategy (combining hard negative sample mining and dynamic negative sample pool updating) to generate more challenging negative samples in the comparison learning process, thus improving the model's discriminative ability; at the same time, it introduces a negative attention mechanism in the graph encoding stage, which combines dynamically screened difficult negative samples, and by dynamically adjusting the weight of the negative samples in the attention aggregation, it further suppresses the noise interference and strengthens the key information extraction. The experimental results show that CLNSO significantly outperforms the existing methods in terms of AUC, F1, and Recall@ on two public datasets, Book-Crossing and MovieLens-1M, which verifies its effectiveness in dealing with the problem of data sparsity and noise.**

Index Terms—**Recommendation, Negative sample optimization, Contrastive learning, Knowledge graphs**

I. Introduction

As a core technology in the field of information filtering, recommender systems play an irreplaceable role in e-commerce, social media, and content distribution platforms.

Its core objective is to filter personalized content from massive information by analyzing users' historical behaviors and preferences, so as to improve user experience and enhance the commercial value of the platform. However, with the exponential growth of data size, traditional recommendation methods gradually expose two major bottlenecks: first, the data sparsity [1] problem is prevalent, especially in the long-tail user and cold item scenarios, where more than 90% of the entries in the user-item interaction matrix are missing, which makes it difficult for traditional methods, such as Collaborative Filtering (CF), to capture effective association signals; second, the cold start problem severely restricts the utility of the system, as the embedded representations of new users or items cannot be accurately initialized due to the lack of historical interaction data, which in turn affects the reliability of the recommendation results. To address the above challenges, researchers have turned to Knowledge Graph (KG) to introduce rich semantic information. Knowledge graphs construct multidimensional associations between entities (e.g., "movie-director-Spielberg," "user-purchase-merchandise") through structured triples (head entity-relationship-tail entity), which are capable of revealing the underlying patterns of users' interests at the semantic level. However, traditional knowledge graph embedding methods (e.g., TransE [2] and TransH [3]) can only model simple first-order linear relationships (e.g., pairs of directly connected entities), ignoring the higher-order interaction semantics among entities (e.g., users indirectly associate movie genres or production companies through their favorite directors). This limitation leads to the diversity and accuracy of recommendation results that are difficult to meet the practical needs, especially when dealing with complex long-tail distribution.

In recent years, Knowledge-aware Recommendation (KGR [4]) has gradually become a research hotspot in academia and industry, and its core idea is to build a more discriminative recommendation model by fusing the structured semantics of knowledge graphs and user behavior data. The current research on knowledge-aware recommender systems mainly focuses on three directions: knowledge-aware recommendation methods, comparative learning, and attention mechanism optimization. Embedding-based approaches (e.g., PGACKG [5] and SAID [6]) map entities to low-dimensional vectors through knowledge graph embedding techniques, but their linear assumptions and static processing make it difficult to capture dynamic user preferences. MetaKG [7] utilizes a meta-learning framework to address cold-start problems in knowledge graphs; the RecDCL [8]framework optimizes user and item representations through batch comparison learning and feature comparison learning. Graph Neural Networks (GNN) based methods (e.g., KGAT [9] and KGIN [10]) enhance the representation through multi-hop neighbor aggregation but fail to effectively distinguish between local interactions and global associations; the AKUPP [11]framework proposed by Ma et al. combines user preference propagation

Manuscript received July 10, 2025; revised July 12, 2025. This work was supported byNational Natural Science Foundation of China (62272093), Social ScienceFederation project of Liaoning Province (2025lslybwzzkt-100), andPostgraduate education and teaching reform research project of LiaoningProvince(LNYJG2024092).

Jie Yang is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (email: 2635664087@qq.com).

Wenhua Cui is a professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (corresponding author to provide phone: +86-133-0422-4928; e-mail:taibeijack@126.com).

Ye Tao is an associate professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning,Anshan, China (e-mail: taibeijack@163.com).

Tianwei Shi is an associate professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: tianweiabbcc@163.com).

and knowledge graph learning to solve the problem of higher-order relationship mining and data sparsity by reinforcing important relationships through multi-layer propagation and attention mechanisms. Contrastive learning methods (e.g., KGCL [12], MCCLK [13]) utilize self-supervised signals to alleviate the data sparsity problem but generally suffer from insufficient quality of negative samples, and randomized or regularized negative sampling strategies are difficult to provide consistently effective training signals. The KGSL [14]framework introduces knowledge graphs for enriching the item representations and devises cross-view contrastive learning mechanisms to deal with noisy data. In terms of the attention mechanism, although the existing models (e.g., MBGCN [15] and SMIN [16]) can dynamically allocate aggregation weights, they lack the effective use of negative sample information and are unable to achieve the synergistic optimization of noise suppression and comparative learning objectives; KGNN-LS [17]adjusts the attention weights through label smoothing techniques, but it relies on manually setting the smoothing parameters. These limitations lead to the existing methods still having significant deficiencies in negative sample quality, dynamic noise suppression, etc., which restricts the performance improvement of recommender systems.

In order to break through the above bottleneck, this paper proposes a knowledge-aware recommendation algorithm based on comparative learning and negative sample optimization (CLNSO). The core contributions of the model are reflected in the following three aspects: 1) An innovative dynamic negative sample selection strategy is proposed, which effectively solves the problem of insufficient quality of negative samples in traditional comparative learning by filtering the high-difficulty negative samples that are semantically similar to the positive samples but have not interacted with them through the hard-negative sample mining technique and combining with the adaptive updating mechanism to continually optimize the pool of negative samples. 2) A negative attention mechanism is designed, which combines the difficult negative samples of dynamic screening in the process of graph encoding and adjusts the attention weights through the similarity of negative samples, which significantly improves the model's suppression ability of noise interference and the extraction effect of key features. 3) Extensive experiments on two baseline datasets, Book-Crossing and MovieLens-1M, show that the proposed method significantly outperforms the existing baseline methods in key metrics such as accuracy, recall, and F1 value, verifying the effectiveness and superiority of the model.

The paper is organized as follows: Section 2 outlines the related work. Section 3 identifies the problem to be addressed by the study. Section 4 describes in detail the model proposed in this paper. Section 5 describes the experimental design and evaluation. Section 6 summarizes the work of this paper and discusses future research directions.

II. Related Work

This chapter introduces the core theoretical and technological foundations on which the methodology of this paper relies, including knowledge graph representation learning, graph neural networks, comparative learning frameworks, and their applications in recommender systems, to provide theoretical support for subsequent model design.

A. Knowledge Graph Representation Learning

Knowledge graphs represent semantic relationships between entities through the ternary (h, r, t) where h is the head entity, r is the relationship, and t is the tail entity. Knowledge graph representation learning aims to map entities and relations to a low-dimensional continuous vector space, preserving their semantic associations. Classical approaches include:1) translational models (e.g., TransE): assuming that the relation vector r is approximated as the difference between the head and tail entity vectors, i.e., $h + r \approx t$, which is suitable for modeling simple first-order relations; and 2) rotational models (e.g., RotatE [18]): mapping entity embeddings into a complex vector space and modeling complex relations (e.g., symmetry, inverse relations) by complex multiplications.

Although KGE can effectively capture direct associations between entities, its ability to model higher-order semantics (e.g., multi-hop paths) and nonlinear relationships is limited and needs to be further optimized in conjunction with graph neural networks.

B. Graph Neural Networks (GNNs)

The GNN iteratively updates the node representation by aggregating the local neighborhood information of the nodes through a message-passing mechanism. The formal expression of its core operation is shown in Equation (1).

$$h_v^{(1)} = \text{AGGREGATE}\left(\{h_u^{(1-1)} \mid u \in N(v)\}\right). \tag{1}$$

where $N(v)$ denotes the set of neighbors of node v and $AGGREGATE$ is the aggregation function (e.g., mean pooling, attention weighting). In knowledge-aware recommendation, GNN is used to fuse user-item interaction graphs with knowledge graphs, but traditional methods (e.g., KGAT) are prone to introducing noise because they do not distinguish between local and global semantics.

C. Contrastive Learning Framework

Contrastive learning drives the model to learn discriminative representations by maximizing the similarity of positive sample pairs and minimizing the similarity of negative sample pairs. Its objective function is usually based on InfoNCE loss, which takes the form shown in Equation (2).

$$L_{\text{contrast}} = -\log \frac{\exp(s(z_i, z_j)/\tau)}{\sum_{k=1}^{N} \exp(s(z_i, z_k)/\tau)}. \tag{2}$$

where $s(\cdot)$ is the similarity function (e.g., cosine similarity) and τ is the temperature coefficient. In recommender systems, contrastive learning is used to mitigate data sparsity (e.g., KGCL), but its negative sample selection strategy (e.g., random sampling) is prone to introducing low-quality training signals and limiting model performance.

D. Knowledge-aware Recommender Systems

Knowledge-aware recommender systems improve recommendation performance by integrating user behavior data and knowledge graph semantic information. According to the different ways of knowledge fusion, the existing methods can be mainly classified into two categories: embedding fusion and graph structure modeling. Embedding fusion methods such as CKE [19] enrich the representation by splicing or weighted combinations of knowledge graph entity embeddings and user-item embeddings; graph structure modeling methods such as KGIN construct user-item-entity heterogeneous information networks and utilize graph neural networks for end-to-end semantic propagation and learning. Although these methods have achieved good results in specific scenarios, they generally ignore the critical impact of negative sample quality on model performance and do not make full use of negative sample information to optimize the attention mechanism, and these shortcomings provide important improvement directions for the research work in this paper.

Although existing methods (e.g., KGIC [20]) improve recommendation performance by fusing knowledge graphs and user behavior, their limitations in negative sample selection and noise suppression remain unresolved; the traditional negative sampling strategy is difficult to provide highly discriminative training signals, and the attention mechanism lacks dynamic utilization of negative sample information. To address these issues, this paper proposes a novel knowledge-aware recommendation framework in Chapter 3, which achieves more robust semantic representation learning by co-optimizing the negative sample selection and attention aggregation mechanisms.

III. METHODOLOGY

A. Definition Of The Problem

Two core types of structured data are first defined: user-item interaction data and knowledge graphs, and knowledge-enhanced recommendation tasks are formally described.

In a recommender system, let $u = \{u_1, u_2, \cdots, u_M\}$ denote the set containing M users and $v = \{v_1, v_2, \cdots, v_N\}$ denote the set containing N items. The interaction between a user and an item is represented by the user-item interaction matrix $Y \in R^{M \times N}$, where $y_{uv} = 1$ denotes that there is an interaction between user u and item v, and $y_{uv} = 0$ denotes that there is no record of interaction between the two.

In order to incorporate more external information in the recommender system, a knowledge graph G, represented by the set of triples $G = \{(h, r, t) \mid h \in \varepsilon, r \in R, t \in \varepsilon\}$, is introduced, where h and t are the head entity and the tail entity, respectively; r denotes the relationship between them, and ε and R are the set of entities and the set of relationships, respectively. In the recommendation task, some of the items $v \in V$ are mapped to entities $e \in \varepsilon$ in the knowledge graph. For this purpose, the model constructs the set of item-entity alignments $A = \{(v, e) \mid v \in V, e \in \varepsilon\}$, where (v, e) denotes the item v corresponding to the entity e in the knowledge graph. With this alignment relation, the recommender system is able to combine user-item interaction data with knowledge graph information to generate more accurate and semantically rich recommendation representations.

B. Overview Of The CLNSO Model

Among the existing knowledge-aware recommender systems, the KGIC [20]model has made significant progress in improving recommendation performance by combining knowledge graph and contrast learning techniques. The core innovation of the model lies in the construction of a dual graph structure of local and non-local graphs and and the adoption of a multilevel comparative learning framework within and between graphs. Specifically, KGIC first captures the direct interaction information between users and items through local graphs while mining cross-domain higher-order semantic relationships using non-local graphs; then it realizes the alignment of collaborative filtering signals with knowledge graph information in intra-graph comparative learning and establishes the association between local and non-local graphs in inter-graph comparative learning. This design effectively mitigates the data sparsity problem and enhances the representation capability of the model.

However, after in-depth analysis, it is found that there are still two key deficiencies in the KGIC model: firstly, in terms of negative sample selection, KGIC adopts a static negative sample pool and a fixed-rule sampling strategy, which is unable to dynamically identify the “difficult negative samples” that are semantically similar to the positive samples, resulting in a lack of sufficient discriminative training signals during the comparative learning process; secondly, in the stage of information aggregation, the attention mechanism in the Secondly, in the information aggregation stage, the attention mechanism of KGIC fails to fully consider the interference effect of negative samples, which makes the noise information in the knowledge graph over-amplified in the propagation process and affects the extraction effect of key features.

To address the above problems, this paper proposes the CLNSO model shown in Fig. 1, which is systematically improved on the basis of the KGIC framework. In terms of negative sample optimization, CLNSO innovatively designs a dynamic negative sample selection strategy (shown in the blue dashed box in Fig. 1), which significantly improves the training effect of comparison learning through the hard negative sample mining and adaptive updating mechanism; in terms of graph coding, CLNSO proposes a negative attention mechanism (shown in the purple dashed box in Fig. 2), which combines the dynamically screened difficult negative samples with negative sample similarity to adjust the attention weights, effectively suppressing noise interference and enhancing the extraction of key features. These improvements allow CLNSO to further enhance the robustness of the model in data sparse and noisy interference scenarios while maintaining the original advantages of KGIC.

Specifically, the improvement of CLNSO is mainly reflected in the following three aspects: first, the negative sample quality problem caused by static sampling of KGIC is solved by the dynamic negative sample selection strategy; second, the negative attention mechanism is introduced to optimize the noise filtering ability of KGIC in the process of information aggregation; and third, the semantic consistency between the local graphs and the non-local graphs is enhanced by the joint optimization framework.

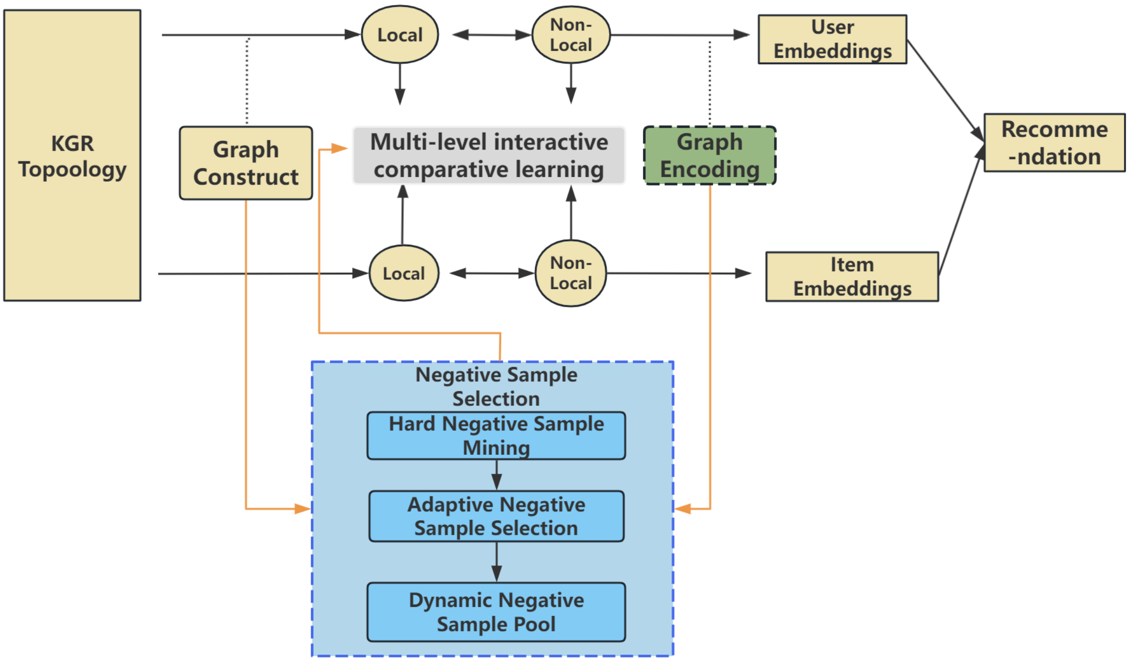
KGR Topoology
Graph Construct
Local
Non-Local
Multi-level interactive comparative learning
Graph Encoding
Local
Non-Local
User Embeddings
Item Embeddings
Recomme -ndation
Negative Sample Selection
Hard Negative Sample Mining
Adaptive Negative Sample Selection
Dynamic Negative Sample Pool

Fig. 1. CLNSO modeling framework

C. Map Encoding

KGIC uses an ordinary attention mechanism for graph encoding, aggregating tail entity embeddings by computing the attention weights $\pi(e_{\mathrm{h}}, r)$ of head entities and relations. However, KGIC does not consider negative sample interference, which results in noisy nodes (e.g., irrelevant entities) being given higher weights in information propagation, affecting key feature extraction. For this reason, CLNSO proposes the negative attention mechanism (Eqs. 7, 22, 25), which introduces a negative similarity penalty term in the attention weight computation, which dynamically reduces the weights of the neighbors that are semantically unrelated to the current node, thus suppressing noise propagation.

The graph encoding phase maps the multi-level information in the local and non-local graphs into low-dimensional embeddings to generate the final representation of the user and the project, which consists of the following four main modules, as shown in Figure 2.

(1) Mapping user IDs, item IDs and entities and relations in KG are mapped as low-dimensional dense vectors to generate the user/item initial embedding and KG entity/relationship embedding. The embedding representation is calculated as shown in Eqs. (3)-(6).

$$E_u = \text{Embedding}_{\text{user}}(\text{id}_u), \tag{3}$$

$$E_i = \text{Embedding}_{\text{item}}(\text{id}_i), \tag{4}$$

$$E_e = \text{Embedding}_{\text{entity}}(\text{id}_e), \tag{5}$$

$$E_r = \text{Embedding}_{\text{relation}}(\text{id}_r). \tag{6}$$

where E_u and E_i represent the embedding representations of users and items, respectively, and E_e and E_r represent the entity and relationship embeddings in the knowledge graph. Since simple embedding is not enough to capture complex inter-entity relationships, to further enhance the information aggregation effect, the model introduces a negative attention mechanism in the attention aggregation layer.

(2) Calculate the attention weights of the target node and neighbor nodes as shown in equation (7).

$$\alpha_{(e,e')} = \alpha_{(e,e')} - \lambda \cdot \text{NegAtt}(E_e, E_{e'}), \tag{7}$$

where λ is the moderating parameter for negative attention and NegAtt$(\cdot)$ is used as a function to measure negative affect (based on similarity).

After obtaining the adjusted attentional weights, the information of all neighboring nodes is then weighted and summed using Eq. (8).

$$E_e^{\text{agg}} = \sum_{e' \in N(e)} \alpha_{(e,e')} E_{e'}, \tag{8}$$

where $E_{e'}$ denotes the embedded representation of the neighboring entity e', $\alpha_{(e,e')}$ is the attention weight adjusted by the negative attention mechanism, and $N(e)$ denotes the set of neighbors of the target entity e. The process generates an aggregation vector E_e^{agg}.

The original embedding of the target entity is combined with the neighbor information obtained through the aggregation of Eq. (8) to generate the final entity embedding representation shown in Eq. (9).

$$E_e^{(\text{final})} = \text{MLP}(E_e + E_e^{\text{agg}}). \tag{9}$$

where E_e^{agg} is the neighbor information aggregation vector, E_e denotes the original embedding of the target entity e, MLP$(\cdot)$ denotes the multilayer perceptron, and E_e is the vector spliced with the neighbor information E_e^{agg} obtained from the aggregation as an input to generate the final representation $E_e^{(\text{final})}$ through a nonlinear transformation.

(3) The model splices the original embedding E_u of the target node with the aggregated neighbor context vector E_e^{final} as shown in Equation (10).

$$E_u^{(\text{final})} = \text{concat}(E_u + E_e^{\text{final}}), \tag{10}$$

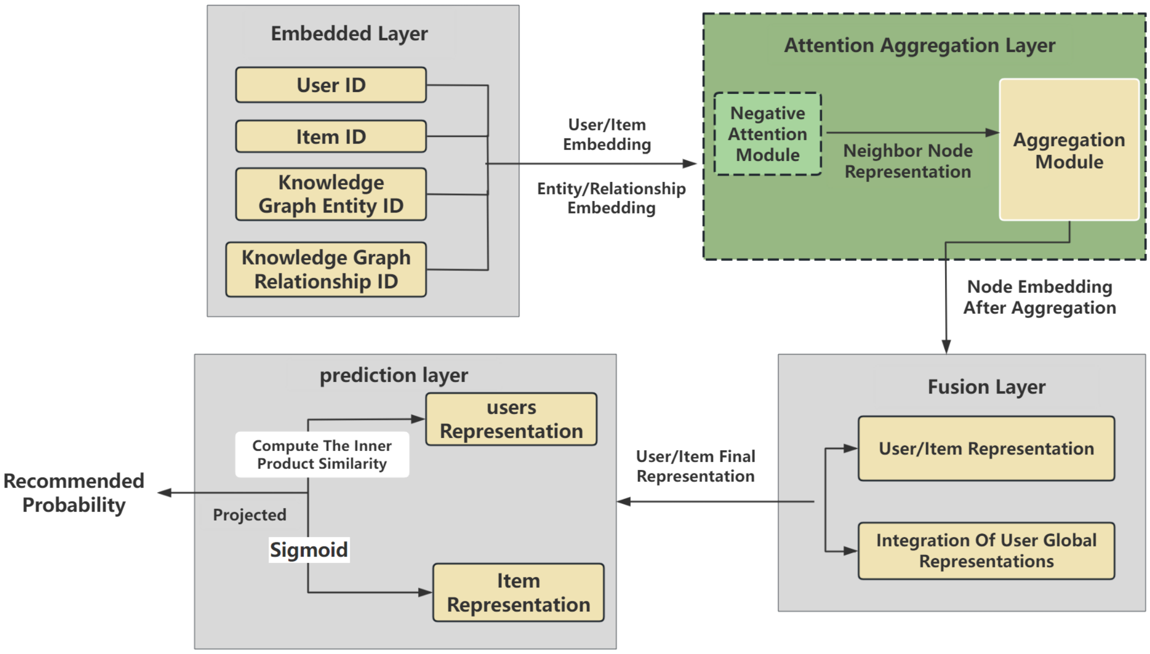
Embedded Layer
User ID
Item ID
Knowledge Graph Entity ID
Knowledge Graph Relationship ID
User/Item Embedding
Entity/Relationship Embedding
Attention Aggregation Layer
Negative Attention Module
Neighbor Node Representation
Aggregation Module
Node Embedding After Aggregation
Fusion Layer
User/Item Representation
Integration Of User Global Representations
User/Item Final Representation
prediction layer
users Representation
Compute The Inner Product Similarity
Projected
Sigmoid
Item Representation
Recommended Probability

Fig. 2. Diagram coding framework

where E_u denotes the original embedding of the target node and E_e^{final} is the neighbor information obtained through the aggregation of Eq. (9).

Subsequently, in order to further extract the higher-order features in the joint vectors and map them to the final representation space, the model inputs $E_u^{(\text{final})}$ into a multilayer perceptron MLP and performs a nonlinear transformation to obtain the final fused representation shown in Eq. (11).

$$E_u^{(\text{fusion})} = \text{MLP}(E_u^{(\text{final})}), \tag{11}$$

where $E_u^{(\text{fusion})}$ combines the embedded information of the user itself and the higher-order information learned from the knowledge graph.

(4) In the prediction stage, the model firstly utilizes Eq. (12) to compute the inner product of E_u and E_i to obtain the matching score $S(u, i)$ between the user and the item, which reflects the similarity between the user and the item in the embedding space, i.e., the higher the similarity, the larger the matching score.

$$S_{(u,i)} = E_u^{fusion} \cdot E_i, \tag{12}$$

$$p_{(u,i)} = \sigma(S_{(u,i)}). \tag{13}$$

Next, the model uses the $sigmoid$ function in Eq. (13) to normalize the matching scores by mapping them to the 0,1 interval, which yields the final recommendation probability $p(u, i)$.

D. Negative Sample Selection Strategy

KGIC uses a static pool of negative samples, which is generated by random sampling. This strategy does not dynamically adapt to the training process, resulting in a large number of “simple negative samples” (which are significantly different from the positive samples) being selected, weakening the discriminative power of comparative learning. To this end, a dynamic negative sample selection strategy is designed in the CLNSO framework to optimize the training signals for contrast learning through hard negative samples and adaptive updating.

(1) The model screens the negative samples that are most similar to the positive samples based on cosine similarity in each training cycle. Let the positive sample embedding be E_p, the candidate negative sample embedding be E_n, and the set of candidate negative samples be N; the cosine similarity is calculated as shown in equation (14).

$$S(E_p, E_n) = \frac{E_p \cdot E_n}{\|E_p\|\|E_n\|}, \tag{14}$$

where E_p and E_n are vector representations of positive and negative samples, respectively. The higher $S(E_p, E_n)$ is, the closer the semantics of the negative sample are to the positive sample.

Based on the cosine similarity score, the k negative samples with the highest similarity are selected as the hardest negative samples. The specific process is shown in Eqs. (15) and (16).

$$N_{hard} = \text{arg}_{E_n \in N}^{max} S(E_p, E_n), \tag{15}$$

$$\|N_{hard}\| = k, \tag{16}$$

Since the parameters of the model are constantly updated during training, the difficulty of the negative samples will also change. Therefore, we recalculate and update the pool of negative samples at the end of each training, as shown in Eq. (17).

$$N_{pool} \leftarrow \text{Update}(N_{pool}, N_{hard}). \tag{17}$$

At the end of each training, the currently selected set of hardest negative samples N_{hard} is dynamically updated to the pool of negative samples N_{pool} through the Update$(\cdot)$ function.

(2) To further enhance the effect of contrast learning, the model uses InfoNCE loss to measure the differentiation

between positive and negative samples, as shown in Equation (18).

$$L_{\text{contrastive}} = -\log \frac{\exp(s(E_p, E_p^+))}{\sum_{E_n \in N} \exp(s(E_p, E_n))}, \tag{18}$$

where E_p^+ is the true pairwise term (positive example) for the positive sample, $S(E_p, E_p^+)$ represents the similarity of the positive sample pair, and $S(E_p, E_n)$ represents the similarity of the positive sample to the negative sample.

Next, in order to adjust the pool of negative samples based on the contrast loss feedback, the loss value of each candidate negative sample L_n is calculated as shown in Equation (19).

$$L_n = -\log \frac{\exp(s(E_p, E_n))}{\sum_{E_k \in N} \exp(s(E_p, E_k))}, \tag{19}$$

where L_n is used to measure the "difficulty" of comparison learning for each negative sample E_n, the larger the loss value L_n, the more similar E_n is to the positive sample E_p.

Subsequently, as shown in Eq. (20), k negative samples with higher losses are selected from the candidate negative samples to form the set of adaptive negative samples.

$$|N_{\text{adaptive}}| = k. \tag{20}$$

This ensures that the model always learns the most challenging negative samples throughout the training process.

E. Comparative Learning

KGIC learns to align local and non-local graph representations through inter-graph comparison, but its comparison objective relies only on ordinary similarity computation and does not jointly optimize negative sample selection and attention mechanisms. In addition, KGIC's non-local graphs are susceptible to higher-order noise (e.g., irrelevant cross-domain entities), resulting in limited semantic alignment effects. To this end, the joint optimization framework CLNSO is proposed to combine the negative attention mechanism with dynamic negative sample selection for inter-graph comparison: 1) Cross-graph negative attention (Eq. 25): the negative similarity penalty is reused in the attention aggregation of non-local graphs to suppress the noise node interference. 2) Dynamic negative sample support: negative samples used in the inter-graph comparison are also from the dynamic pool, ensuring that the comparison task focuses on difficult samples.

Based on graph coding, a dual optimization framework for local and non-local contrast learning is further proposed, and the model is able to extract more discriminative user-item representations from local direct interaction information and non-local latent knowledge.

(1) Local contrast learning uses a negative attention mechanism to filter hard negative samples in the local graph and optimize the user embedding representation. During the training process, the model needs to maximize the similarity $S(E_u, E_v)$ between the user embedding E_u and the positive sample embedding E_v while suppressing the similarity with the hard negative sample E_n, as shown in Equation (21).

$$S(E_u, E_v) \gg S(E_u, E_n), \forall n \in N_{hard}, \tag{21}$$

where N_{hard} is the set of hardest negative samples.

The negative attention mechanism is introduced in the attention aggregation layer to dynamically adjust the neighbor node weights as shown in Equation (22).

$$\alpha_{(e,e')}^{\text{local}} = \alpha_{(e,e')} - \lambda \cdot \text{NegAtt}(E_e, E_{e'}), \tag{22}$$

where e' is a neighboring entity in the local graph and $\text{NegAtt}(E_e, E_{e'}) = S(E_e, E_{e'})$ measures negative sample similarity.

To achieve this goal, InfoNCE-based contrast loss function is used as shown in Eq. (23).

$$L_{\text{local}} = -\log \frac{\exp(s(E_u, E_p^+)/\tau)}{\sum_{E_n^- \in N_i} \exp(s(E_u, E_n^-)/\tau)}. \tag{23}$$

where E_p^+ represents all localized positive sample pairs (user-items), N_i is the set of negative samples, selected from the pool of negative samples, and τ is the temperature parameter, used to adjust the smoothing of the loss.

(2) Non-local contrast learning utilizes non-local graphs to capture latent interest and cross-layer semantic consistency in order to compensate for the inadequacy of local graphs in long-range dependency modeling. To maintain consistency of information across layers, it is assumed that the representation of the same entity in successive layers should be as similar as possible, as shown in Equation (24).

$$E^{(l)} \approx E^{(l+1)}, \tag{24}$$

In the attentional aggregation of nonlocal graphs, Eq. (7) is reused, but the range of the adjustment weight calculation is shown in Eq. (25).

$$\alpha_{(e,e')}^{\text{non-local}} = \alpha_{(e,e')} - \lambda \cdot \text{NegAtt}(E_e, E_{e'}), \tag{25}$$

where e' is the higher - order neighboring entities in the nonlocal graph, while NegAtt computes the adaptation cross - domain similarity.

Based on the above adjustments, the non-localized contrast loss is defined as shown in Eq. (26).

$$L_{\text{non-local}} = -\sum_{(i,j)\in p} \log \frac{\exp(s(E_u^{(l)}, E_p^{(l+1)})/\tau)}{\sum_k \exp(s(E_u^{(l)}, E_k)/\tau)}, \tag{26}$$

where $E_u^{(l)}$ is the entity embedding in layer l and $E_p^{(l+1)}$ is the embedding in layer $l+1$.

In addition, in order to keep the negative sample pool in the nonlocal graph always contains the most indistinguishable samples, this study prioritizes the negative samples that contribute the most to $L_{non-local}$ based on the contrast loss feedback and updates the negative sample pool as shown in Eqs. (27) and (28).

$$N_{hard}^{\text{non-local}} \leftarrow \text{arg}_{E_n^- \in N}^{max} \left(-\log \frac{\exp(S(E_u, E_n^-))}{\sum \exp(S(E_u, E_k^-))} \right), \tag{27}$$

$$L_{\text{total}} = L_{\text{local}} + L_{\text{non-local}}. \tag{28}$$

Combining the localized contrast loss and the non-localized contrast loss gives the final contrast learning loss L_{total}.

TABLE I
DATA SET STATISTICS AND HYPERPARAMETER SETTINGS

Component	Variant	Book-Crossing	MovieLens-1M
User - item Interaction	# users	17,860	6,036
	# items	14,967	2,445
	# interactions	139,746	753,772
knowledge Graph	# entities	77,903	182,011
	# relations	25	12
	# triplets	151,500	1,241,996
Hyperparameter Settings	# lr	0.001	0.001
	# λ_1	1×10^{-6}	1×10^{-6}
	# d	16	16
	# t	0.2	0.2
	# λ_2	1×10^{-4}	1×10^{-4}

(lr: learning rate, λ_1: weight of loss, d: Embedding dimensions, t: temp, λ_2: regularization)

IV. EXPERIMENTS

To answer the following research questions, we conducted extensive experiments on two public datasets.

Question 1 (RQ1): What is the performance of the CLNSO model compared to existing models under different datasets?

Question 2 (RQ2): Do the main components (e.g., negative attention mechanisms, negative sample selection strategies) actually work well?

Question 3 (RQ3): How do different hyperparameter settings affect the CLNSO model?

A. Experimental Setup

1) Description Of The Data Set: Two publicly available data sets were used to evaluate the effectiveness of CLNSO: Book-Crossing and MovieLens-1M. They vary in size and sparsity to make the model more convincing. The basic statistics of the two datasets are shown in Table 1.

Book-Crossing: a book dataset from the Book-Crossing community, containing user ratings of books (ranging from 0 to 10), covering a wide range of book genres and user behavior data, providing a rich resource for book recommendations.

MovieLens-1M: contains about 1 million movie ratings data, which are in the range of 0 to 5, reflecting users' intuitive evaluation of movies.

To ensure data consistency, RippleNet [21] was followed to convert explicit feedback to implicit feedback in both datasets, where Table 1 represents the positive samples. In order to determine the optimal number of negative samples K, this study used a grid search (candidate range: 5, 10, 20, 50) to select the value K with the highest AUC on the validation set to ensure the validity of negative sample selection. Ultimately, each positive sample matches K negative samples, i.e., a $1 : K$ sampling strategy is used.

In the construction process of subknowledge graphs, the methods of RippleNet and KGCN [22] are referred to and constructed using Microsoft Satori knowledge graphs. Specifically, each subknowledge graph is stored in the form of a ternary, and a subset with a confidence level higher than 0.9 is filtered from the whole knowledge graph. After construction, all valid movie/book IDs are collected by matching the entity names at the end of the triples and further matching the heads of the triples corresponding to these IDs so as to filter out the compliant triples and finally construct the sub-knowledge graphs. Table 1 shows the basic statistical information of the two datasets.

In this study, the dataset is divided according to the ratio of 6:2:2, in which 60% is used as the training set for model training, 20% as the validation set for hyperparameter tuning, and 20% as the test set for final performance evaluation. The data are divided by random sampling, i.e., 20% of the data are randomly selected as the validation set, 20% of the remaining data are selected as the test set, and the remaining 60% are used as the training set, in order to ensure a balanced distribution of the data and the stability of the experimental results.

2) Evaluation Indicators: In order to fully evaluate the performance of this model, this study validates it in two experimental scenarios:

(1) In the click-through rate prediction (CTR) task, AUC and F1 scores are used as evaluation metrics to measure the discriminative ability and predictive effectiveness of the model, as shown in Eqs. (29) and (30).

$$AUC = \frac{1}{|P| \cdot |N|} \sum_{p \in P} \sum_{n \in N} 1(s_p > s_n). \tag{29}$$

where$P$ and N denote the set of positive and negative samples, respectively; S_p and S_n are the predicted scores of the positive and negative samples, respectively; while $1(\cdot)$ is the indicator function (taking 1 if the condition in parentheses holds, 0 otherwise).

(2) In the Top-K recommendation task, Recall@K is used as an evaluation criterion, where K is set to 5, 10, 20, 50, and 100 to measure the recall of recommendation results.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{30}$$

In this case, the precision and recall rates are calculated as shown in equations (31) and (32).

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{31}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{32}$$

TP, FP, and FN represent the number of true, false positive, and false negative cases, respectively.

In the Top-K recommendation task, Recall@K is used as an evaluation criterion to measure the proportion of the first K results of the recommendation list that contain positive samples, as shown in Equation (33).

$$\text{Recall@}K = \frac{|R \cap T_K|}{|R|}. \tag{33}$$

$|R \cap T_K|$ denotes the number of correctly recommended items in the intersection of the set of the user's real relevant items R and the first K items T_k of the recommendation list; $|R|$ denotes the total number of all real related items of the user.

3) Comparison of Methods: In order to demonstrate the effectiveness of the proposed CLNSO, CLNSO is compared with state-of-the-art methods, including traditional recommendation algorithms based on LibFM, SVD, and BPRMF; embedding-based methods (CKE, RippleNet, SHINE); path-based methods (PER); and GNN-based methods (CG-KGR, KGIN, MCCLK, KGIC, LightGCN, DKN, KGCN, KGNN-LS, CKAN) as follows:

LibFM [23]: Combining feature engineering with decomposition modeling for learning via stochastic gradient descent with Markov Chain Monte Carlo inference.

SVD [24]: Fusion of neighborhood and latent factor models, integration of explicit and implicit feedback.

BPRMF [25]: Optimizing personalized ranking by stochastic gradient descent with a self-sampling algorithm based on Bayesian optimization criterion.

CKE [19]: Combining collaborative filtering with structural, textual, and visual knowledge representations in knowledge graphs for joint learning of multimodal information.

RippleNet [21]: Iteratively expands user interests through ripple propagation generates a preference distribution of candidate items.

SHINE [26]: A multi-view self-encoder embedding model is used to fuse emotional, social, and user profile information to effectively deal with cold start problems.

PER [27]: Diffuses user preferences in heterogeneous information networks using meta-paths and achieves high-quality recommendations through Bayesian ranking optimization.

CG-KGR [28]: A personalized recommendation model based on a knowledge graph that extracts user and external knowledge information through a co-guidance mechanism.

KGIN [10]: Modeling user intent and relationship path perception to capture fine-grained relationships between users and items.

MCCLK [13]: Multi-view comparison learning model to enhance semantic feature extraction by comparing local and global graph views.

KGIC [20]: It is a knowledge graph-based recommendation model that unifies collaborative filtering and knowledge graph information by constructing local and non-local maps and learning from multi-level interactive comparisons within and between graphs.

LightGCN [29]: Simplified version of graphical convolutional networks, retaining only the core components of neighborhood aggregation for efficient propagation of user-item embeddings.

DKN [30]: Combining knowledge graph with CNN for semantic representation in news recommendation and capturing user interests through attention mechanism.

KGCN [22]: Based on graph convolutional networks, higher-order structural and semantic features are captured by aggregating entity neighborhood information.

KGNN-LS [17]: The model transforms the knowledge graph into an exclusively weighted graph via a user-specific relational scoring function, and combines it with label smoothing regularization for the end-to-end learning of personalized item embeddings for recommendation.

CKAN [31]: The model is a recommender system that fuses collaborative filtering with knowledge graphs to capture knowledge associations in user-item interactions through an attention mechanism.

4) Parameter Setting: In this study, the Adam optimizer is used for model training. In order to improve the model's performance, a grid search is conducted for the key hyperparameter of negative sampling size. The candidate values are [5, 10, 20, 50], and the optimal negative sampling size is finally determined to be 20. Meanwhile, the batch size is set to be 64, and the sizes of the local ternary set and the non-local ternary set are 40 and 128, respectively. In addition, to adequately capture the higher-order neighbor information, the depth of the graph neural network is set to be 2 layers. The model CLNSO framework is implemented using PyTorch, and the specific configurations of other hyperparameters are detailed in Table 1, and these settings are referred to as the empirical values from related literature and previous studies.

B. Comparative Test (RQ1)

The model CLNSO is experimentally compared with a variety of baseline models on two public datasets, Book-Crossing and MovieLens-1M. The results of the experiments are shown in Tables 2 and 3, with the best results in bold and the second-best results underlined. In terms of overall performance, this model shows strong competitiveness in both AUC and F1 evaluation metrics, which fully demonstrates its effectiveness in knowledge-aware recommendation tasks.

TABLE II
COMPARATIVE EXPERIMENTAL RESULTS UNDER THE BOOK-CROSSING DATASET

Model	AUC	F1
LibFM[23]	0.691	0.618
SVD[24]	0.672	0.635
CKE[19]	0.6759	0.6235
RippleNet[21]	0.7211	0.6472
CG-KGR[28]	0.7578	0.6714
KGIN[10]	0.7273	0.6614
MCCLK[13]	0.7625	0.6777
KGIC[20]	0.7464	0.6676
LightGCN[29]	0.6134	0.6469
CKAN[31]	0.7420	0.6671
KGNN-LS[17]	0.6762	0.6314
KGCN[22]	0.6841	0.6313
CLNSO	**0.7725**	**0.6818**

TABLE III
COMPARISON EXPERIMENT RESULTS UNDER MOVIELENS-1M DATASET

Model	AUC	F1
BPRMF[25]	0.8920	0.7921
CKE[19]	**0.9065**	0.8024
SHINE[26]	0.778	–
PER[27]	0.7124	0.6670
KGIC[20]	0.8834	0.8056
LightGCN[29]	0.8880	0.8091
DKN[30]	0.655	–
KGCN[22]	0.8918	0.8166
CLNSO	0.8880	**0.8171**

On the Book-Crossing dataset, this model achieves 0.7725 and 0.6818 on the AUC and F1 metrics, respectively, and shows a significant advantage over other baseline models. Specifically, the performance of traditional matrix decomposition methods (e.g., SVD) is relatively limited, and both AUC and F1 are at a low level, making it difficult to effectively capture the complex relationship between users and items. In contrast, the knowledge graph enhancement models (e.g., CKE, SHINE, PER, KGIC, LightGCN) outperform the traditional methods as a whole, among which the AUC and F1 of MCCLK are already close to the optimal. However, compared with these models, the present model further improves the differentiation ability of representation learning after introducing dynamic negative sample selection and a negative attention mechanism and ultimately outperforms the other methods in both AUC and F1 metrics, reflecting a more accurate portrayal of users' implicit preferences.

In the MovieLens-1M dataset, this model also shows good performance: the AUC metric reaches 0.8880, which is comparable to excellent baseline models (e.g., LightGCN); and the F1 metric achieves 0.8171, which exceeds a variety of knowledge-graph-augmented or GNN models, including KGIC, KGCN, etc., which indicates that this model performs excellently in both recommendation accuracy and robustness, indicating that this model has excellent performance in recommendation accuracy and robustness. Other models, such as BPRMF, PER, etc., have significantly lower metrics on this dataset, showing the limitations of traditional recommendation models in large-scale data and high sparsity scenarios.

C. Ablation Studies (RQ2)

In order to validate the roles of the negative sample selection strategy and the negative attention mechanism in the overall performance of the model, this study designs ablation experiments that assess their impact by removing these two modules separately and comparing them with the full model CLNSO.

First, to verify the role of the dynamic negative sample selection strategy in the model, version CLNSO$_{w/o\ \text{Non-local}}$ is constructed, i.e., dynamic negative sample selection is removed and replaced by random negative sample sampling. In this version, the model no longer uses the hard sample mining with adaptive negative sample selection mechanism in the negative sample generation process but uses the traditional random sampling method. This approach cannot guarantee the quality of negative samples, and it is difficult to generate difficult negative samples with high similarity to positive samples, thus weakening the differentiation ability of the model.

Secondly, in order to verify the role of the negative attention mechanism in the model, version CLNSO$_{w/o\ \text{inter}}$ is constructed, i.e., the negative attention mechanism is removed and is replaced by normal attention aggregation. In this version, the model no longer suppresses irrelevant or noisy information during neighbor information aggregation but gives the same weight to all neighbors, thus weakening the model's ability in information filtering and feature extraction, and the experimental results are shown in Table 3.

Model CLNSO$_{w/o\ \text{Non-local}}$: Due to random negative sampling, it is not possible to generate more challenging, high-quality negative samples, and the model's representational learning ability is reduced and differentiation is weakened.

Model CLNSO$_{w/o\ \text{inter}}$: Due to the lack of a negative attention mechanism, the model is susceptible to irrelevant or noisy feature interference when aggregating neighbor information, leading to the weakened expression of useful features and ultimately affecting the overall performance.

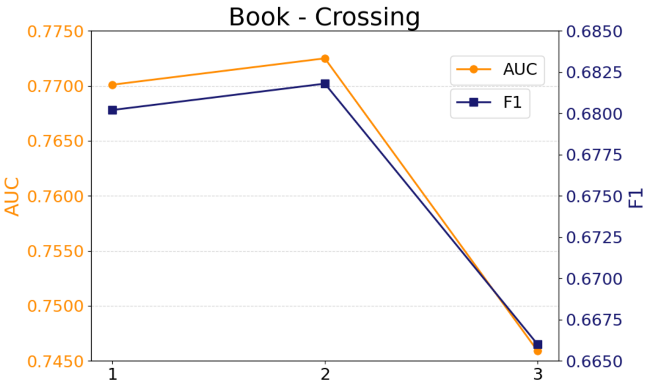

Fig. 3. Comparative experimental results of Book-Crossing

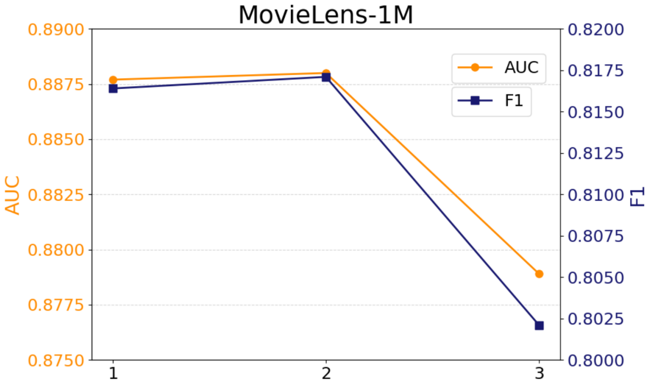

Fig. 4. Comparative experimental results of MovieLens-1M

The experimental results show that the dynamic negative sample selection strategy can effectively generate more challenging negative samples and enhance the differentiation ability of the model. Meanwhile, the negative attention mechanism can effectively suppress irrelevant or noisy information and strengthen the representation ability of the model. The two complement each other and jointly enhance the overall performance of the model.

D. Parameterization Experiment (RQ3)

Finally, we investigated the effect of aggregation layers on the model performance. Experiments were conducted to adjust the number of network layers in the range of 1, 2, 3 and the results are shown in Figures 3 and 4. On the Book-Crossing and MovieLens-1 datasets, the AUC and F1 scores are optimized when the model depth is two layers. As the model depth increases, the model is able to capture higher-order neighborhood information, which enhances the representation learning capability. However, too many graph propagation layers may lead to an oversmoothing problem that converges the node embeddings and weakens the discriminative ability of the model.

V. CONCLUSION

Aiming at the shortcomings of existing recommendation models in dealing with data sparsity and noise interference, this study proposes the CLNSO model, which effectively improves the discriminative ability and robustness of the model by introducing a dynamic negative sample selection strategy and a negative attention mechanism. On the MovieLens-1M

TABLE IV
RESULTS OF ABLATION EXPERIMENTS.

Method	MovieLens-1M		Book-Crossing	
	AUC	F1	AUC	F1
CLNSO$_{w/o \text{ Non-local}}$	0.7464	0.6676	0.8834	0.8056
CLNSO$_{w/o \text{ inter}}$	0.7560	0.6708	0.8864	0.8159
$CLNSO$	0.7725	0.6818	0.8880	0.8171

dataset, the improved model improves the AUC by about 0.5% and the F1 score by 1.4%, while on the Book-Crossing dataset, the AUC and F1 are improved by about 3.5% and 2.1%, respectively. These results show that CLNSO has significant advantages in capturing users' potential interests and fusing higher-order knowledge information. Overall, the method in this paper not only optimizes the integration of knowledge graph and collaborative filtering information but also effectively mitigates the data sparsity and noise problems through the dual optimization of comparative learning, providing new ideas and strong support for the development of knowledge-aware recommender systems.

DATA AVAILABILITY STATEMENT

The datasets used in this article are publicly available data resources from the following sources:

MovieLens 1M dataset is available through the official website provided by GroupLens Research: https://grouplens.org/datasets/movielens/1m/.

Book- The Crossing dataset is provided by Universität Freiburg and is available at the following link: http://www.informatik.uni-freiburg.de/~cziegler/BX/.

REFERENCES

[1] J. Wu et al., "Self-supervised graph learning for recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 726-735.

[2] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data," *Advances in neural information processing systems*, vol. 26, 2013.

[3] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014.

[4] Y. Liu, W. Wei, A. Sun, and C. Miao, "Exploiting geographical neighborhood characteristics for location recommendation," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 739-748.

[5] Y. Li, L. Hou, and J. Li, "Preference-aware graph attention networks for cross-domain recommendations with collaborative knowledge graph," *ACM Transactions on Information Systems*, vol. 41, no. 3, pp. 1-26, 2023.

[6] J. Hu et al., "Enhancing sequential recommendation via llm-based semantic embedding learning," in *Companion Proceedings of the ACM Web Conference 2024*, 2024, pp. 103-111.

[7] Y. Du, X. Zhu, L. Chen, Z. Fang, and Y. Gao, "Metakg: Meta-learning on knowledge graph for cold-start recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 9850-9863, 2022.

[8] D. Zhang et al., "Recdcl: Dual contrastive learning for recommendation," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 3655-3666.

[9] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & data mining*, 2019, pp. 950-958.

[10] X. Wang et al., "Learning intents behind interactions with knowledge graph for recommendation," in *Proceedings of the Web Conference 2021*, 2021, pp. 878-887.

[11] X. Ma, L. Dong, Y. Wang, Y. Li, and H. Zhang, "Akupp: Attention-enhanced joint propagation of knowledge and user preference for recommendation systems," *Knowledge and Information Systems*, vol. 65, no. 1, pp. 163-182, 2023.

[12] Y. Yang, C. Huang, L. Xia, and C. Li, "Knowledge graph contrastive learning for recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1434-1443.

[13] D. Zou et al., "Multi-level cross-view contrastive learning for knowledge-aware recommender system," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1358-1368.

[14] S. Li, Y. Jia, Y. Wu, N. Wei, L. Zhang, and J. Guo, "Knowledge-Aware Graph Self-Supervised Learning for Recommendation," *Electronics*, vol. 12, no. 23, p. 4869, 2023.

[15] B. Jin, C. Gao, X. He, D. Jin, and Y. Li, "Multi-behavior recommendation with graph convolutional networks," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 659-668.

[16] X. Long et al., "Social recommendation with self-supervised metagraph informax network," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1160-1169.

[17] H. Wang et al., "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 968-977.

[18] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[19] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 353-362.

[20] D. Zou et al., "Improving knowledge-aware recommendation with multi-level interactive contrastive learning," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 2817-2826.

[21] H. Wang et al., "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 417-426.

[22] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The World Wide Web Conference*, 2019, pp. 3307-3313.

[23] S. Rendle, "Factorization machines with libfm," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, pp. 1-22, 2012.

[24] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 426-434.

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

[26] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "Shine: Signed heterogeneous information network embedding for sentiment link prediction," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 592-600.

[27] X. Yu et al., "Personalized entity recommendation: a heterogeneous information network approach," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, 2014, pp. 283-292.

[28] Y. Chen, Y. Yang, Y. Wang, J. Bai, X. Song, and I. King, "Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation," in *Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2022, pp. 299-311.

[29] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn:

Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639-648.

[30] H. Wang, F. Zhang, X. Xie, and M. Guo, "Dkn: Deep knowledge-aware network for news recommendation," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1835-1844.

[31] Z. Wang, G. Lin, H. Tan, Q. Chen, and X. Liu, "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*, 2020, pp. 219-228.