DKGFN: Deep Knowledge Graph Fusion Network for Recommendation

Jibing Gong[1,2], Le Chen[1], Qian Zang[1], Xinchao Feng[1], Yi Zhao[3*], Yu Qiu[4], Jun Feng[5], Haomin Song[5], Liangying Peng[5]

[1]School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei 066004, China

[2]Research Institute of Yanshan University, Shenzhen Virtual University Park, Shenzhen, Guangdong 518063, China

[3]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

[4]Zhipu AI, Beijing 100085, China

[5]Artificial Intelligence Innovation Center, State Grid Zhejiang Electric Power Corporation Information & Communication Branch, Hangzhou, Zhejiang 310014, China

Keywords: Knowledge-aware recommendation; Contrastive learning; Dual-view attention

Citation: Gong J.B., Chen L., Zang Q., et al.: DKGFN: Deep Knowledge Graph Fusion Network for Recommendation. Data Intelligence, 2026, Vol. 8, Art. No.: 20250100. DOI: https://doi.org/10.3724/2096-7004.di.2025.0100

ABSTRACT

Knowledge graphs (KG) are rich in semantic entities and multi-hop relational structures, which can effectively capture implicit associations among item attributes. In contrast, user-item interaction graphs reflect explicit behavioral patterns between users and items. Integrating these two types of graphs enables the discovery of more complex and fine-grained high-order correlations between user preferences and item semantics, thereby enhancing the expressive power of representation learning and improving the accuracy of personalized recommendations. However, existing methods often lack deep integration between knowledge graphs and user-item interaction graphs. To address this limitation, we propose a deep dynamic fusion framework named DKGFN, which leverages layer-wise contrastive learning and a dual-view attention mechanism to achieve deep fusion of the two graph modalities. Experimental results demonstrate that DKGFN outperforms strong baselines by 8.3% and 11.2% in terms of Recall and NDCG, respectively, validating the effectiveness of our approach.

1. INTRODUCTION

With the growing challenge of information overload, recommender systems have emerged as essential tools for helping users discover relevant content [1]. Collaborative Filtering (CF) methods, based on the

*Corresponding author: Yi Zhao (E-mail: zhaoyi.hb@gmail.com; ORCID: 0000-0002-0919-867X).

© 2026 Chinese Academy of Sciences. Published under a Creative Commons Attribution 4.0 International (CC BY 4.0) license.

assumption that "similar users prefer similar items" have achieved strong performance across various domains [2-3]. However, a persistent challenge in CF-based methods is the data sparsity problem, which significantly hinders accurate user preference modeling, especially for cold-start users and items.

Recent advances in KG-enhanced recommendation have leveraged the rich semantics of knowledge graphs to alleviate data sparsity and improve recommendation performance. Some approaches incorporate semantic-aware item representations or multi-hop relational reasoning via Graph Neural Networks (GNNs) [4-9]. While these methods achieve notable gains, they often emphasize the KG structure while only superficially integrating signals from the user-item interaction graph. Consequently, they fail to achieve deep fusion between the two heterogeneous views, which limits the full potential of joint modeling.

Beyond this limitation, existing models frequently adopt static or early-fusion strategies for combining KG and CF signals [4, 10-11], which can introduce redundant or noisy representations. A further challenge arises from the long-tail distribution and representation degradation during graph aggregation. In real-world data, a small fraction of popular items and active users account for most interactions, while the majority in the long tail receive very few, as shown in Figure 1. Similarly, entity connectivity in the KG is also highly imbalanced, as shown in Figure 2. Multi-layer aggregation in GNNs tends to cause over-smoothing, making long-tail node representations indistinguishable from those of high-frequency nodes and thereby weakening their expressiveness. Meanwhile, insufficient handling of heterogeneous relations restricts the model's ability to capture diverse signals.

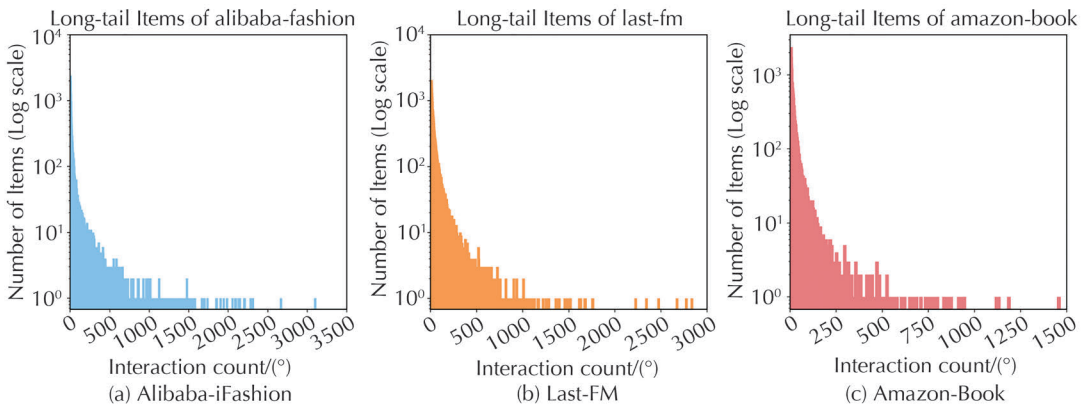

Figure 1. Long-tail Item Distribution in Interaction Data.

To address the above challenge, we propose a Deep Knowledge Graph Fusion Network Recommendation Framework named DKGFN which integrates three key components: **i) Dual-view Attention Mechanism**: We introduce a dynamic attention module to adaptively adjust the weights of collaborative and knowledge signals during fusion. This mechanism highlights informative signals while suppressing noisy or redundant ones, enabling accurate alignment of user preferences. **ii) Layer-wise Contrastive Learning**: We design a layer-wise contrastive learning strategy by injecting perturbations into representations at each layer and

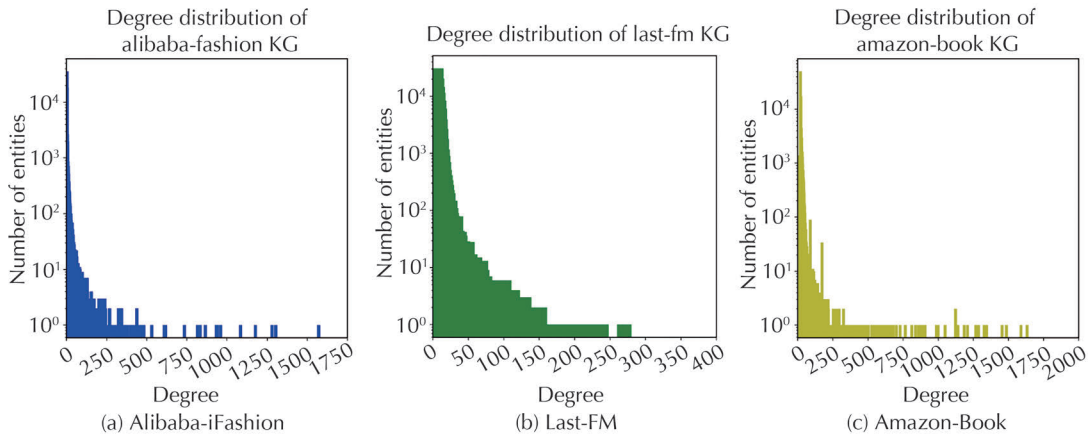

Figure 2. Long-tail Entity Distribution in the Knowledge Graph.

optimizing them through contrastive objectives. This improves the inter-layer discriminability of node embeddings and alleviates over-smoothing in GNNs. **iii) View-wise Contrastive Learning**: We construct an auxiliary graph-level contrastive task to align the distributional structures of the collaborative and knowledge views. This global alignment enhances the overall fusion of collaborative and knowledge signals and improves both model robustness and recommendation performance.

In summary, our main contributions are as follows:

- We propose a novel dual-view attention mechanism and a layer-wise contrastive learning strategy to better integrate signals from collaborative and knowledge graphs, while maintaining sufficient distinction between node representations across layers.
- We introduce a view-wise contrastive learning task to align the signal distributions of different views, which improves the model's ability to capture user preferences and item attributes from a macro-level perspective.
- Extensive experiments on three real-world datasets validate the effectiveness of our proposed DKGFN model. The evaluation results provide strong evidence that DKGFN significantly outperforms existing knowledge-aware recommendation baselines.

2. RELATED WORK

2.1 Knowledge Graph-Based Recommendation

As an important external source of information for modeling user preferences and item semantics, knowledge graphs have demonstrated significant potential in enhancing the performance of recommender systems. Existing knowledge-aware recommendation approaches can be broadly categorized into three technical paradigms: embedding-based methods, path-based methods, and graph neural network-based methods.

Embedding-based methods enhance the representational capacity of collaborative filtering by integrating representations of entities or relations from the knowledge graph into user and item embeddings. For example, CKE [12] combines modeling of multiple types of side information with collaborative filtering by employing TransR [13] to encode entity relations as part of item representations. Path-based methods focus on mining semantic paths within the knowledge graph. A representative work, KPRN [14], utilizes LSTM [15] to model extracted meta-paths and aggregates users' preferences over these paths through fully connected layers, thereby improving recommendation accuracy and interpretability.

In recent years, GNN-based methods have gained popularity due to their capability to aggregate multi-hop information on graph structures, effectively integrating the strengths of both embedding-based and path-based modeling. KGCN [4] aggregates multi-hop neighbor information via sampling. KGAT [5] applies attention to weigh neighbor importance. KGIN [6] incorporates relation embeddings to capture diverse user preferences. KGRec [16] combines generative and contrastive self-supervised learning. While effective, these methods typically rely on static fusion of collaborative and KG signals, and seldom address view misalignment or representation degradation. In contrast, DKGFN introduces dual-view attention for adaptive fusion, and contrastive strategies at both layer and view levels to enhance robustness and cross-view consistency.

2.2 Contrastive Learning for Recommendation

Contrastive learning, as an effective self-supervised paradigm, initially achieved remarkable success in computer vision [17], and has recently been widely applied in recommender systems [18] to alleviate data sparsity and enhance model robustness. Classical approaches include SGL [19], which generates multiple views via graph structure perturbation, SimGCL [20], which injects noise in the embedding space for contrastive learning, and CCDR [21], which directly constructs contrastive views using dual-domain data in cross-domain recommendation.

For knowledge graph-enhanced recommendation, recent studies have explored leveraging knowledge graphs as auxiliary views for contrastive learning [18], focusing on multi-view construction and semantic alignment mechanisms. For example, KGIC [22] performs hierarchical contrastive learning via subgraph sampling. MCCLK [23] constructs three types of views for both local and global representation learning. KGCL [11] conducts multi-scale contrastive learning based on random augmentation. KACL [24] optimizes a dual-objective framework through edge dropout between the user-item graph and the KG. KGRec [16] fuses generative and contrastive signals to filter irrelevant noise. KGTN [25] introduces intention-aware contrastive denoising, and DCLKR [26] enhances multi-dimensional representation capabilities through both intra-view and inter-view contrastive learning. These studies collectively demonstrate the effectiveness of contrastive learning in improving recommendation accuracy and robustness.

3. PROBLEM DEFINITION

Suppose the recommendation system contains M users and N items. The user set and item set are denoted as $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$, respectively. The user-item interactions can be represented as a binary interaction matrix $Y \in \mathbb{R}^{M \times N}$, where $y_{ui} = 1$ indicates that user u has interacted with item i (e.g., click, purchase, or watch), and $y_{ui} = 0$ otherwise. Based on the interaction matrix, a bipartite collaborative filtering graph is constructed as:

$$G_{\mathrm{CF}} = \{(u, y_{ui}, i) | u \in \mathcal{U}, i \in \mathcal{I}\} \tag{1}$$

On the other hand, the knowledge graph on the item side provides rich semantic information such as directors, actors, and genres. It is formalized as a set of knowledge triples:

$$G_{\mathrm{KG}}^{raw} = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\} \tag{2}$$

Here, $\mathcal{E}$ denotes the set of entities, and $\mathcal{R}$ is the set of relation types. Each triple describes a semantic relation r between the head entity h and the tail entity t.

Inspired by the modeling strategy in KGAT [5], we further unify the interaction graph and the knowledge graph into a single collaborative knowledge graph (CKG, hereinafter referred to as KG):

$$G_{\mathrm{KG}} = \{(h, r, t) | h, t \in \mathcal{E}', r \in \mathcal{R}'\} \tag{3}$$

where $\mathcal{E}' = \mathcal{E} \cup \mathcal{U}$ and $\mathcal{R}' = \mathcal{R} \cup \{\mathrm{Interact}\}$. This unified graph structure allows the model to integrate multi-source signals within a consistent graph representation framework.

The architecture of the proposed DKGFN model is illustrated in Figure 3. It consists of four core modules: (i) the graph embedding module, (ii) the dual-view attention fusion module, (iii) the layer-wise perturbation contrastive learning module, and (iv) the view-wise contrastive learning module.

4. METHODOLOGY

4.1 Graph Embedding Module

To address potential interference between knowledge and collaborative signals, we adopt a **dual-channel strategy** to propagate information independently in the knowledge and collaborative graphs.

Knowledge-graph View. To better capture the dynamic influence of different relation types on entity representations, we introduce a Heterogeneous Graph Transformer. During each layer of message passing, the model dynamically adjusts the contribution of various relations to entity embeddings using relation-specific projection matrices and attention mechanisms. The item embedding update for item i in the knowledge view at layer $l + 1$ is defined as:

$$e_i^{\mathrm{KG}(l+1)} = \frac{1}{|N_i|} \sum_{(i, r, v) \in N_i} \alpha(v, r, i) \cdot e_r \odot e_v^{(l)} \tag{4}$$

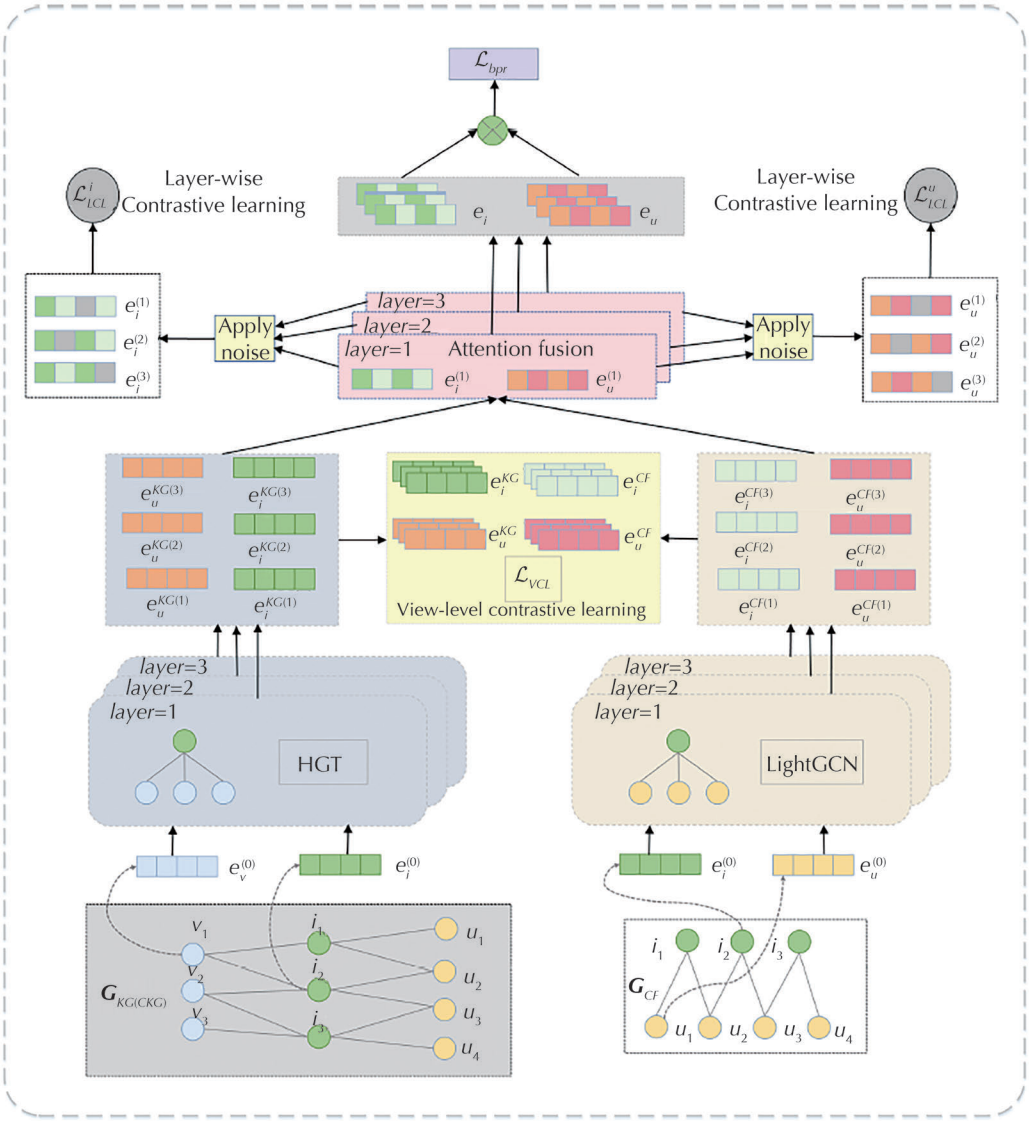

Figure 3. DKGFN Framework.

The attention weight $\alpha(v, r, i)$ is computed as:

$$\alpha(v,\ r,\ i) = \text{Softmax}\left(\frac{(e_v W_Q)\cdot(e_r \odot e_i W_K)^\top}{\sqrt{d}}\right) \tag{5}$$

where W_Q and W_K are learnable projection matrices, and $\sqrt{d}$ is a scaling factor. The Hadamard product between relation r and entity v is used to model relational context and adaptively emphasize different relations in varying semantic spaces.

Collaborative View. To capture collaborative signals efficiently while keeping the model lightweight, we adopt the LightGCN [27] architecture for the collaborative view. The user embedding update at layer $l + 1$ is defined as:

$$e_u^{\mathrm{CF}(l+1)} = \sum_{i \in N_u} \frac{e_i^{\mathrm{CF}(l)}}{\sqrt{|N_u| \cdot |N_i|}} \tag{6}$$

the item embedding $e_i^{\mathrm{CF}(l+1)}$ update is computed similarly.

4.2 Dual-view Attention Fusion Module

To mitigate interference between knowledge-graph and the collaborative view, DKGFN adopts a fine-grained signal-aware attention mechanism with an aggregate-then-fusion strategy. At each propagation layer, attention weights are dynamically computed to adaptively fuse the two signals. This enables precise balancing of both views and reduces noise and information loss from naive aggregation.

The fused item embedding for item i at layer $l + 1$ is computed as:

$$e_i^{(l+1)} = \omega(e_i^{\mathrm{KG}(l+1)}, e_i^{(l)}) \cdot e_i^{\mathrm{KG}(l+1)} + \omega(e_i^{\mathrm{CF}(l+1)}, e_i^{(l)}) \cdot e_i^{\mathrm{CF}(l+1)} \tag{7}$$

The signal attention weights are computed using dot-product attention as:

$$\omega(e_i^{\mathrm{KG}(l+1)}, e_i^{(l)}) = \frac{\exp(e_i^{(l)} \cdot e_i^{\mathrm{KG}(l+1)})}{\sum_{s \in \{\mathrm{KG}, \mathrm{CF}\}} \exp(e_i^{(l)} \cdot e_i^{s(l+1)})} \tag{8}$$

$$\omega(e_i^{\mathrm{CF}(l+1)}, e_i^{(l)}) = \frac{\exp(e_i^{(l)} \cdot e_i^{\mathrm{CF}(l+1)})}{\sum_{s \in \{\mathrm{KG}, \mathrm{CF}\}} \exp(e_i^{(l)} \cdot e_i^{s(l+1)})} \tag{9}$$

Similarly, the fused embedding for user $e_u^{(l+1)}$ at layer $l + 1$ is computed similarly.

After obtaining the embeddings from all propagation layers, i.e., $e_i^{(0)}, e_i^{(1)}, \ldots, e_i^{(l)}$ and $e_u^{(0)}, e_u^{(1)}, \ldots, e_u^{(l)}$, we aggregate them using a sum aggregation function to obtain the final representations:

$$e_i = \mathrm{AGG}\{e_i^{(0)}, e_i^{(1)}, \ldots, e_i^{(l)}\}, \; e_u = \mathrm{AGG}\{e_u^{(0)}, e_u^{(1)}, \ldots, e_u^{(l)}\} \tag{10}$$

Here, AGG{·} denotes the sum aggregation operation.

4.3 Layer-wise Contrastive Learning Module

The long-tail distribution in interaction and knowledge data often leads to representation degradation and over-smoothing, reducing recommendation accuracy. To address this, we introduce layer-wise contrastive learning to enhance the expressiveness of user and item embeddings.

At each GCN layer l, we generate two perturbed views of the item embedding $e_i^{(l)}$ by applying masking noise and Gaussian noise, enabling contrastive learning across diverse representations.

The first augmented view is generated by applying Bernoulli masking noise to the item embedding, as shown in Equation 11:

$$e_i^{(l,\,1)} = e_i^{(l)} \odot M, \quad M \sim \text{Bernoulli}(1-p) \tag{11}$$

Here, M is a random mask sampled from a Bernoulli distribution, and p controls the masking probability, determining the strength of the noise.

Inspired by the data augmentation strategy in SimGCL [20], we generate the second augmented view by injecting Gaussian noise into $e_i^{(l)}$, which promotes a more uniform distribution of embeddings in the feature space. The formulation is given in Equation 12:

$$e_i^{(l,\,2)} = e_i^{(l)} + \Delta_i, \quad \| \Delta_i \|_2 = \epsilon, \quad \Delta_i = \bar{\Delta}_i \odot \text{sign}(e_i^{(l)}), \quad \bar{\Delta}_i \sim \mathcal{N}(0,\,1) \tag{12}$$

In this formulation, ϵ controls the magnitude of the perturbation. The noise vector Δ_i is aligned with the sign of the original embedding to ensure the perturbed vector lies in the same quadrant, thereby preserving semantic consistency.

Next, we employ the InfoNCE loss [28] to perform contrastive learning across layers. For each item, representations from different layers and different augmented views are treated as *positive pairs* if they belong to the same item, and as *negative pairs* otherwise. The item-level layer contrastive loss is defined as:

$$\mathcal{L}_{\text{LCL}}^{i} = \frac{1}{L(L-1)} \sum_{l=1}^{L} \sum_{l' > l} -\log \frac{\exp\left(s(e_i^{(l,\,1)},\, e_i^{(l',\,2)}) / \tau\right)}{\sum_{j \in \mathcal{B}} \exp\left(s(e_i^{(l,\,1)},\, e_j^{(l',\,2)}) / \tau\right)} \tag{13}$$

Here, $s(\cdot)$ denotes cosine similarity, τ is the temperature parameter, and $\mathcal{B}$ is the batch of items used for contrastive sampling.

Similarly, we compute the user-level layer contrastive loss $\mathcal{L}_{\text{LCL}}^{u}$ in the same way, which is omitted here for brevity. The final total layer contrastive loss is the sum of the user and item losses:

$$\mathcal{L}_{\text{LCL}} = \mathcal{L}_{\text{LCL}}^{i} + \mathcal{L}_{\text{LCL}}^{u} \tag{14}$$

4.4 View-level Contrastive Learning Module

To further enhance the integration of collaborative and knowledge signals, we introduce a view-level contrastive learning objective that globally aligns the node representation distributions from the two views.

We denote the layer-wise representations of user and item nodes as $e_x^{v(l)}$, where $x \in \{u, i\}$ indicates the node type (user or item), $v \in \{\text{CF}, \text{KG}\}$ indicates the view (collaborative or knowledge), and l denotes the GNN layer index.

The final aggregated representation for each node is obtained via a layer aggregation function (e.g., mean or weighted sum) over all layers:

$$e_x^v = \text{AGG}\left(\left\{e_x^{v(0)}, e_x^{v(1)}, \ldots, e_x^{v(L)}\right\}\right), \quad v \in \{\text{CF}, \text{KG}\}, \quad x \in \{u, i\} \tag{15}$$

Cross-view, different-node pairs: **A**: (e_i^{KG} vs. e_j^{CF}), **B**: (e_i^{CF} vs. e_j^{KG}). Cross-view, same-node pairs: **C**: (e_i^{KG} vs. e_i^{CF}), **D**: (e_i^{CF} vs. e_i^{KG}).

We adopt InfoNCE-style contrastive loss. The item-level view contrastive loss is defined as:

$$\mathcal{L}_{\text{VCL}}^i = -\log\frac{\exp\left(s(C)/\tau\right)}{\sum_{i\neq j}\exp\left(s(A)/\tau\right)} - \log\frac{\exp\left(s(D)/\tau\right)}{\sum_{i\neq j}\exp\left(s(B)/\tau\right)} \tag{16}$$

where $s(\cdot)$ denotes cosine similarity, and τ is a temperature parameter controlling the sharpness of the distribution.

Similarly, the user-level view contrastive loss $\mathcal{L}_{\text{VCL}}^u$ is defined in the same manner. The total view contrastive loss is:

$$\mathcal{L}_{\text{VCL}} = \mathcal{L}_{\text{VCL}}^i + \mathcal{L}_{\text{VCL}}^u \tag{17}$$

4.5 Multi-task Joint Optimization

DKGFN employs a multi-task joint optimization strategy that simultaneously optimizes three loss components: the recommendation loss, the layer-wise contrastive loss, and the view-level contrastive loss.

For the recommendation task, we compute the predicted preference score $\hat{y}_{ui}$ between user u and item i using inner product. Then we adopt the Bayesian Personalized Ranking (BPR) loss [29] as the training objective:

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,\, i,\, j)\in\, \mathcal{O}} -\log\sigma(\hat{y}_{ui} - \hat{y}_{uj}) \tag{18}$$

Here, $\mathcal{O} = \{(u, i, j) | (u, i) \in \mathcal{O}^+, (u, j) \in \mathcal{O}^-\}$ is the training set of triplets, where $\mathcal{O}^+$ and $\mathcal{O}^-$ denote the observed (positive) and unobserved (negative) interactions, respectively. The final fused embeddings $\mathbf{e}_u$ and $\mathbf{e}_i$ are obtained from Section 4.2.3.

The total objective function of the model is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{BPR}} + \lambda_1 \mathcal{L}_{\text{LCL}} + \lambda_2 \mathcal{L}_{\text{VCL}} + \lambda_3 \| \Theta \|_2^2 \tag{19}$$

where λ_1, λ_2, and λ_3 are hyperparameters controlling the weights of the layer-wise contrastive loss, view-level contrastive loss, and L2 regularization term respectively. Θ denotes all trainable parameters in the model.

5. EXPERIMENTS

In this section, we conduct comprehensive experiments on three real-world datasets to address the following research questions (**RQs**) regarding the proposed DKGFN framework:

RQ1: Does DKGFN outperform state-of-the-art baseline models of varying architectures in recommendation performance?

RQ2: How do core components of DKGFN contribute to its overall efficacy?

RQ3: How sensitive is DKGFN to hyperparameter configurations?

5.1 Experiment Settings

5.1.1 Datasets

To validate the effectiveness of Layer Contrastive Learning and Signal Attention (DKGFN), we employ three public benchmark datasets: Alibaba-Fashion, Last-FM, and Amazon-Book for evaluating DKGFN's performance in Top-K recommendation tasks.

Detailed statistical information of the processed datasets for DKGFN experiments is presented in Table 1.

Table 1. Statistics of Three Evaluation Datasets.

Statistics	Alibaba-iFashion	Last-FM	Amazon-Book
# Users	114,737	23,566	70,679
# Items	30,040	48,123	24,915
# Interactions	1,781,093	3,034,796	847,733
# Density	5.2e-4	2.7e-3	4.8e-4
Knowledge Graph			
# Entities	59,156	58,266	88,572
# Relations	51	9	39
# Triplets	279,155	464,567	2,557,746

5.1.2 Evaluation Protocols

For Top-K recommendation, which generates ranked lists of top-K relevant items, we adopt Recall@K and Normalized Discounted Cumulative Gain (NDCG@K) as primary evaluation metrics. We conduct top-K recommendation experiments on three public datasets: Alibaba-iFashion, Last-FM, and Amazon-Book, using Recall@20 and NDCG@20 as evaluation metrics.

5.2 Baselines

To evaluate the performance of DKGFN on the top-*K* recommendation task, we compare it against 12 representative baseline methods across four categories: methods without knowledge graphs, embedding-based methods, graph neural network-based methods, and contrastive learning-based methods.

1) Methods without Knowledge Graphs.

- **BPRMF** [29]: A classic collaborative filtering model that utilizes only user-item interaction data. It performs pairwise matrix factorization and is optimized using the BPR loss.
- **NFM** [30]: Combines multilayer perceptrons (MLP) with a factorization machine to model both low-order and high-order feature interactions, improving non-linear representation capability.

2) Embedding-based Methods.

- **CKE** [12]: Employs TransR to embed entities in the knowledge graph and incorporates semantic representations into the matrix factorization framework.
- **CFKG** [31]: Models user-item-entity relationships using TransE and treats recommendation as a triple scoring task.

3) Graph Neural Network-based Methods.

- **KGNN-LS** [10]: Introduces label smoothness regularization to guide GCN aggregation based on user preferences for knowledge triples.
- **KGCN** [4]: Aggregates information from entity neighbors using GNNs to model long-range user interests.
- **KGAT** [5]: Unifies the interaction graph and knowledge graph into a collaborative knowledge graph and uses an attention-based neighbor aggregation mechanism.
- **CKAN** [32]: Uses a heterogeneous propagation strategy to separately encode the interaction and knowledge graphs, and incorporates knowledge-aware attention for signal fusion.
- **KGIN** [6]: Proposes a relation path-aware aggregation mechanism to model user intent as a weighted composition of relations, improving both performance and interpretability.

4) Contrastive Learning-based Methods.

- **KGCL** [12]: Applies a relation-aware graph augmentation strategy and uses cross-view contrastive learning to improve the robustness of user and item representations.
- **MCCLK** [24]: Proposes a multi-level contrastive framework that jointly models local collaborative, local semantic, and global structural views.
- **KGRec** [16]: Unifies generative and contrastive self-supervised learning. It adopts a plausibility-aware masking mechanism to extract high-value knowledge structures relevant to recommendation.

5.2.1 Implementation Details

To ensure fair comparison with baseline methods, all models adopt a unified embedding dimension of 64 initialized via the Xavier method. Hyperparameters for DKGFN were determined through grid search across the following ranges: learning rate {0.1, 0.01, 0.001}, graph convolution layers {1, 2, 3}, layer contrastive learning weight {0.01, 0.05, 0.1, 0.5}, view contrastive learning weight {0.01, 0.03, 0.05, 0.07}, contrastive temperature coefficients {0.2, 0.4, 0.6, 0.8, 1.0}, and node/edge dropout ratios {0.1, 0.3, 0.5, 0.7}. The optimal configuration achieving peak performance on the validation set was utilized for final testing.

5.3 Performance comparison (RQ1)

Table 2 shows the evaluation results for our model and the baselines on three datasets. Overall, DKGFN consistently outperforms all 12 baseline models across both metrics on all three datasets, demonstrating the effectiveness of our model design. Compared with the best-performing baselines, DKGFN improves Recall@20 by 8.4%, 7.1%, and 9.4% on Alibaba-iFashion, Last-FM, and Amazon-Book, respectively. For NDCG@20, the improvements are 9.2%, 11.1%, and 13.2%.

As for **Non-KG Methods (e.g., BPRMF, NFM)**, these models underperform compared to KG-based methods, showing that interaction data alone is insufficient to capture complex semantics. Notably, CFKG performs worse than BPRMF on several datasets, indicating simple embedding methods struggle to utilize KG structure and high-order relations. **GNN Methods (e.g., KGAT, KGIN)** generally outperform embedding-based ones by better modeling structural relations. However, models like KGCN perform poorly on dense datasets (e.g., Amazon-Book) due to noise from abundant KG triples affecting aggregation. **DKGFN vs. Traditional GNNs:** Unlike conventional GNNs that fuse signals after layer aggregation, DKGFN fuses collaborative and knowledge signals before aggregation, dynamically adjusting their importance to reduce noise and improve fine-grained representations. **Contrastive Learning Methods (e.g., KGCL, MCCLK, KGRec)**: These methods benefit from self-supervised learning in KG recommendation. DKGFN extends this by applying layer-wise perturbations and combining layer- and view-level contrastive tasks to enhance representation diversity, reduce over-smoothing, and improve generalization.

Table 2. Performance comparison on three datasets (Recall@20 and NDCG@20). The best and second-best results are marked in bold and underline, respectively.

Model	Alibaba-iFashion		Last-FM		Amazon-Book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPRMF	0.0821	0.0506	0.0834	0.0713	0.1395	0.0736
NFM	0.0798	0.0501	0.0809	0.0684	0.1366	0.0716
CKE	0.0898	0.0622	0.0866	0.0734	0.1168	0.0627
CFKG	0.0668	0.0411	0.0705	0.0622	0.1264	0.0663
KGNN-LS	0.0983	0.0633	0.0880	0.0642	0.1362	0.0560
KGCN	0.0983	0.0633	0.0879	0.0694	0.1111	0.0569
KGAT	0.0979	0.0591	0.0873	0.0744	0.1441	0.0785
CKAN	0.0970	0.0509	0.0812	0.0660	0.1442	0.0698
KGIN	0.1144	0.0723	0.0900	0.0779	0.1281	0.0673
KGCL	0.1146	0.0719	0.0905	0.0769	<u>0.1569</u>	<u>0.0833</u>
MCCLK	0.1089	0.0707	0.0671	0.0603	0.1467	0.0748
KGRec	<u>0.1185</u>	<u>0.0742</u>	<u>0.0910</u>	<u>0.0792</u>	0.1522	0.0807
DKGFN	**0.1284**	**0.0810**	**0.0975**	**0.0880**	**0.1717**	**0.0943**
Gain (%)	+8.4%	+9.2%	+7.1%	+11.1%	+9.4%	+13.2%

5.4 Ablation study (RQ2)

To evaluate the individual contributions of each component in DKGFN, we design five ablation variants and conduct experiments on Alibaba-iFashion, Last-FM, and Amazon-Book. Results are shown in Table 3. Specifically: **–SignalAttn** replaces the signal attention module with simple concatenation. **–LayerCL** and **–ViewCL** remove the layer-wise and view-level contrastive learning modules, respectively. **–KG** and **–CF** use only the knowledge or collaborative signal.

All variants perform worse than the full DKGFN model, confirming the effectiveness of each module. Notably, removing **SignalAttn** causes the largest drop, showing its key role in dynamically balancing collaborative and knowledge signals. **LayerCL** also shows a notable impact, proving useful in mitigating over-smoothing and preserving inter-layer discrimination. While **ViewCL** yields smaller gains, it enhances cross-view consistency. Finally, using only one signal source significantly degrades performance, especially for **–CF**, highlighting the complementary value of knowledge signals in improving semantic understanding.

Table 3. Ablation results of DKGFN with different variants (Recall@20 and NDCG@20). The superscript * denotes the largest change in performance.

Ablation Settings	Alibaba-iFashion		Last-FM		Amazon-Book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
DKGFN	**0.1284**	**0.0810**	**0.0975**	**0.0880**	**0.1717**	**0.0943**
-SignalAttn	0.0999	0.0617	0.0762*	0.0682	0.1302	0.0667
-LayerCL	0.1249	0.0787	0.0856	0.0761	0.1427	0.0750
-ViewCL	0.1251	0.0790	0.0849	0.0750	0.1614	0.0871
-KG	0.1183	0.0743	0.0830	0.0730	0.1315	0.0697
-CF	0.0943*	0.0570*	0.0824	0.0677*	0.1241*	0.0639*

In summary, all modules in DKGFN contribute jointly to improved robustness and generalization.

5.5 Influence of hyperparameters (RQ3)

To assess the sensitivity of the DKGFN model to key hyperparameters, we examine four representative factors: the view-level contrastive loss weight (λ_1), the layer-wise contrastive loss weight (λ_2), the edge dropout rate (ρ_e), and the node dropout rate (ρ_n). Keeping other variables fixed, we analyze their impact on model performance across the three datasets. The results are presented in Figures 4–7.

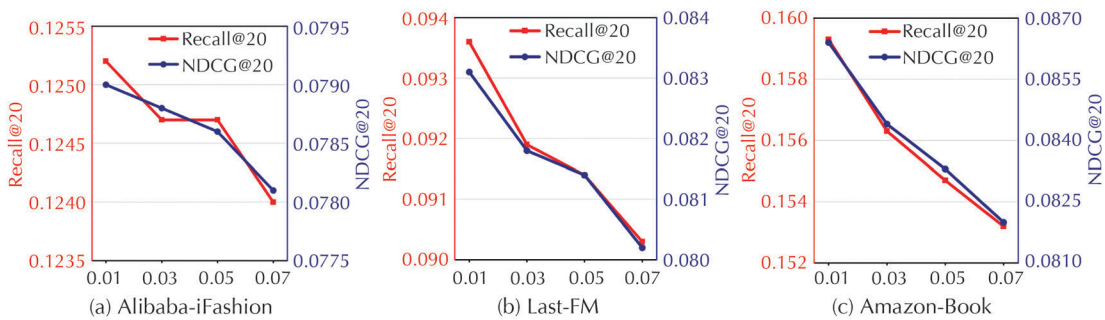

Figure 4. Performance variations of DKGFN under different weights of view-level contrastive loss weight λ_1.

View-level contrastive weight λ_1. As shown in Figure 4, increasing λ_1 from {0.01, 0.03, 0.05, 0.07} consistently degrades model performance across all datasets. This suggests that an excessively strong alignment constraint compresses the representational differences between the collaborative and knowledge views, thereby weakening their complementarity and introducing redundancy. A moderate value of λ_1 strikes a balance by ensuring view consistency while retaining distinctive signals, ultimately enhancing recommendation accuracy.

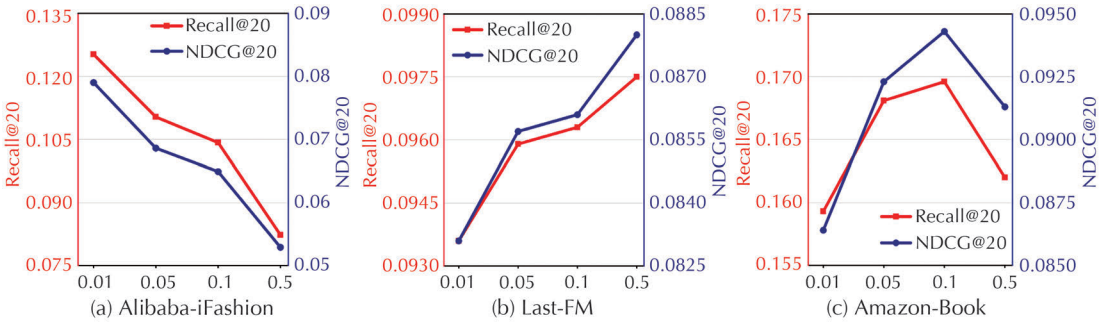

Figure 5. Performance variations of DKGFN under different weights of layer-wise contrastive loss weight λ_2.

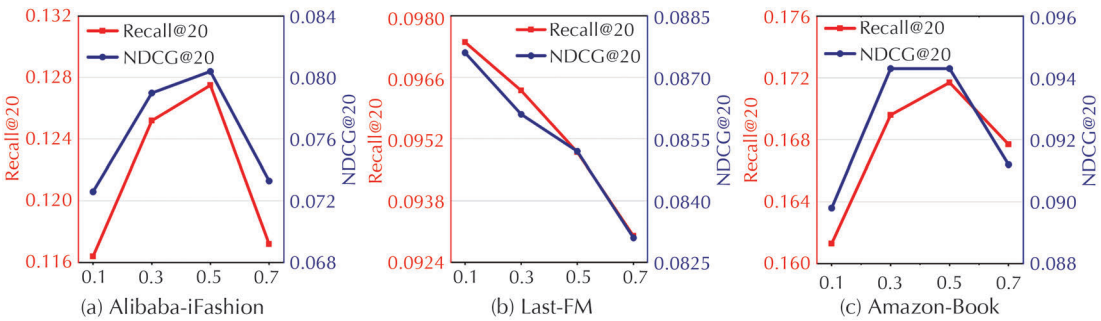

Figure 6. Performance variations of DKGFN under different weights of edge dropout rate ρ_e.

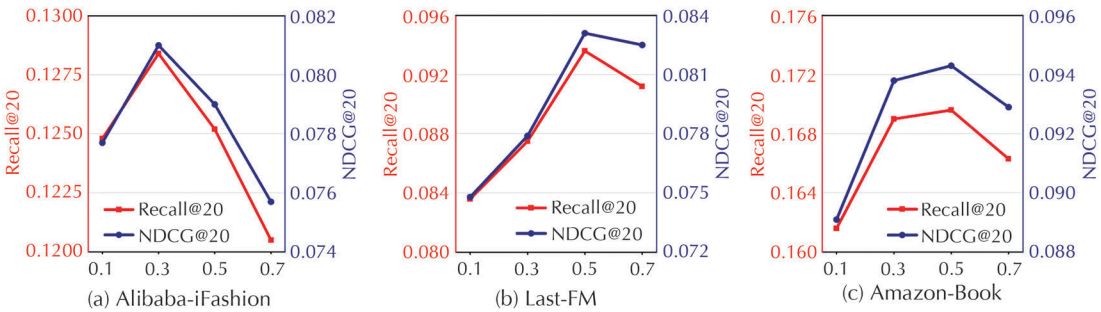

Figure 7. Performance variations of DKGFN under different weights of node dropout rate ρ_n.

Layer-wise contrastive weight λ_2. Figure 5 presents the model performance under $\lambda_2 \in \{0.01, 0.05, 0.1, 0.5\}$. On the Alibaba-iFashion dataset, performance decreases as λ_2 increases, likely due to sparse user–item interactions combined with a dense KG structure, where strong layer-wise contrast introduces noise. In contrast, on Last-FM, performance improves steadily with larger λ_2, suggesting that the mechanism is advantageous in structurally simpler settings. For Amazon-Book, performance peaks at $\lambda_2 = 0.1$ before declining, indicating that overly strong contrastive constraints may hinder optimization of the primary recommendation task. Overall, the effect of λ_2 depends strongly on structural complexity and interaction sparsity.

Edge dropout rate ρ_e. As shown in Figure 6, model performance under $\rho_e \in \{0.1, 0.3, 0.5, 0.7\}$ exhibits distinct trends across datasets. On Last-FM, performance declines steadily as ρ_e increases, suggesting that edge perturbation is particularly detrimental in dense structures. In contrast, on Alibaba-iFashion and Amazon-Book, performance peaks around $\rho_e = 0.5$, indicating that moderate edge dropout can reduce redundancy and mitigate overfitting. Overall, selecting an appropriate ρ_e requires balancing structural complexity against information redundancy.

Node dropout rate ρ_n. Figure 7 illustrates the effect of node dropout rates $\rho_n \in \{0.1, 0.3, 0.5, 0.7\}$. Across all datasets, performance follows a "rise-then-fall" pattern, suggesting that mild perturbation improves robustness, whereas excessive dropout suppresses useful signal propagation. Specifically, Alibaba-iFashion reaches its peak at $\rho_n = 0.3$, while Last-FM and Amazon-Book achieve the best performance at $\rho_n = 0.5$. These results highlight the importance of carefully tuning ρ_n to align with dataset-specific characteristics.

6. CONCLUSION

To overcome the limitations of existing knowledge graph recommendation models, namely insufficient fusion of knowledge and collaborative signals as well as representation over-smoothing and degradation during graph aggregation, this paper introduces DKGFN, a Layer-Contrastive Learning and Dual-view Attention framework. DKGFN incorporates two key innovations: (1) enhanced fusion mechanisms that deeply integrate knowledge and collaborative signals, and (2) a layer-contrastive learning strategy to preserve representation discriminability across layers. The model is optimized through multi-task joint training, and extensive experiments verify its effectiveness, achieving superior performance on Top-K recommendation tasks.

AUTHOR CONTRIBUTIONS

Jibing Gong: Writing review and editing; project administration; supervision; funding acquisition.

Le Chen: Methodology; data curation; validation; writing review and editing; formal analysis, visualization; software; writing original draft.

Qian Zang: Methodology; investigation; validation; writing review and editing; resources.

Xinchao Feng: Visualization; data curation.

Yi Zhao: Writing review and editing; project administration; supervision.

Yu Qiu: Formal analysis; supervision.

Jun Feng: Data curation; writing review and editing.

Haomin Song: Visualization; supervision.

Liangying Peng: Visualization; writing review and editing.

ACKNOWLEDGEMENTS

This work was supported by resources provided by Central Government Guidance Local Science and Technology Development Fund Project (246Z0306G) and Innovation Capability Improvement Plan Project of Hebei Province (22567626H).

REFERENCES

[1] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

[2] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[3] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698, 2018.

[4] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. Knowledge graph convolutional networks for recommender systems. In *The world wide web conference*, pages 3307–3313, 2019.

[5] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019.

[6] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the web conference 2021*, pages 878–887, 2021.

[7] Yi Zhao, Jingxin Ju, Jibing Gong, Jinye Zhao, Mengpan Chen, Le Chen, Xinchao Feng, and Jiquan Peng. Cross-domain recommendation via adaptive bi-directional transfer graph neural networks. *Knowledge and Information Systems*, 67(1):579–602, 2025.

[8] Yi Zhao, Chaozhuo Li, Jiquan Peng, Xiaohan Fang, Feiran Huang, Senzhang Wang, Xing Xie, and Jibing Gong. Beyond the overlapping users: Cross-domain recommendation via adaptive anchor link learning. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1488–1497, 2023.

[9] Jibing Gong, Yi Zhao, Jinye Zhao, Jin Zhang, Guixiang Ma, Shaojie Zheng, Shuying Du, and Jie Tang. Personalized recommendation via inductive spatiotemporal graph neural network. *Pattern Recognition*, 145:109884, 2024.

[10] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 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*, pages 968–977, 2019.

[11] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. Knowledge graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1434–1443, 2022.

[12] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362, 2016.

[13] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

[14] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5329–5336, 2019.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] Yuhao Yang, Chao Huang, Lianghao Xia, and Chunzhen Huang. Knowledge graph self-supervised rationalization for recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3046–3056, 2023.

[17] Zibei Wang. Self-supervised learning in computer vision: A review. In *International Conference on Computer Engineering and Networks*, pages 1112–1121. Springer, 2022.

[18] Mengyuan Jing, Yanmin Zhu, Tianzi Zang, and Ke Wang. Contrastive self-supervised learning in recommender systems: A survey. *ACM Transactions on Information Systems*, 42(2):1–39, 2023.

[19] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021.

[20] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1294–1303, 2022.

[21] Ruobing Xie, Qi Liu, Liangdong Wang, Shukai Liu, Bo Zhang, and Leyu Lin. Contrastive cross-domain recom-mendation in matching. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4226–4236, 2022.

[22] Ding Zou, Wei Wei, Ziyang Wang, Xian-Ling Mao, Feida Zhu, Rui Fang, and Dangyang Chen. Improving knowledge-aware recommendation with multi-level interactive contrastive learning. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 2817–2826, 2022.

[23] Ding Zou, Wei Wei, Xian-Ling Mao, Ziyang Wang, Minghui Qiu, Feida Zhu, and Xin Cao. 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*, pages 1358–1368, 2022.

[24] Hao Wang, Yao Xu, Cheng Yang, Chuan Shi, Xin Li, Ning Guo, and Zhiyuan Liu. Knowledge-adaptive contrastive learning for recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pages 535–543, 2023.

[25] Ding Zou, Wei Wei, Feida Zhu, Chuanyu Xu, Tao Zhang, and Chengfu Huo. Knowledge enhanced multi-intent transformer network for recommendation. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1–9, 2024.

[26] Shuhua Huang, Chenhao Hu, Weiyang Kong, and Yubao Liu. Disentangled contrastive learning for knowledge-aware recommender system. In *International Semantic Web Conference*, pages 140–158. Springer, 2023.

[27] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng 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*, pages 639–648, 2020.

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

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

[31] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9):137, 2018.

[32] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang 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*, pages 219–228, 2020.