Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

MVKGCL: Recommendation Model Based on Knowledge Graph and Contrastive Learning

Zhiyue Xiong[a], Hankiz Yilahun[a,*], Sadiyagul Anwer[b] and Askar Hamdulla[a]

[a]Xinjiang University, Urumqi 830017, China

[b]Xinjiang Agricultural Vocational Technical University, Urumqi 830017, China

ARTICLE INFO

Article History:
Submitted 11.10.2024
Revised 11.25.2024
Accepted 12.5.2024

Keywords:
Knowledge graphs
Recommendation model
Contrastive learning
Graph attention network

A B S T R A C T

Most existing recommendation models based on knowledge graphs and contrastive learning employ random augmentation schemes to enhance the data in knowledge graphs; however, noise in knowledge graphs can lead to inaccurate recommendation results. Furthermore, most contrastive learning methods are only applied between one or two views, thereby failing to exploit the semantic information in the data fully. Therefore, this model proposes a recommendation model, MVKGCL, which integrates knowledge graphs and contrastive learning mechanisms. Firstly, it incorporates random noise into attention weights to conduct contrastive learning among different attention weights, and subsequently introduces a novel automatic masking mechanism to augment the Knowledge Graphs, performing local contrastive learning on the derived user and item embeddings. Secondly, it employs Graph Attention Network to encode the user-item-entity graph, yielding representations for users and items. Lastly, global level contrastive learning is conducted between the locally learned user and item embeddings and the node embeddings from the user-item-entity graph, uncovering comprehensive graph features and structural information. Experiments demonstrate that the model outperforms others on the Amazon-book and Yelp2018 datasets.

1. Introduction

As networks have rapidly advanced and contemporary tech products have become widespread, humanity has entered the epoch of big data, giving rise to colossal volumes of data in everyday life. The capability of users to handle information lags significantly behind the pace at which information disseminates, a dilemma termed the issue of information overload. Recommendation models automatically assist users in pinpointing pertinent information amidst this sea of data, furnishing them with tailored data services. Fundamentally, collaborative filtering recommendation algorithms hinge on scrutinizing user conduct, item characteristics, and the historical interplay between users and items [1,2]. By doing so, they distil the traits of users and items, facilitating individualized recommendations tailored to diverse users. Even though these algorithms endeavor to model intricate dynamics between users and items, numerous models grounded in collaborative filtering grapple with sparse data complications and the cold start predicament. Knowledge graphs (KG), brimming with substantial entities and relational insights, can potentially augment the semantic depictions of both users and items. Consequently, incorporating KG as supplementary data into recommendation frameworks serves as a remedy for these hurdles and bolsters overall efficacy [3]. Nonetheless, KG is beset by issues about noise and a shortage of dense supervisory cues [4].

Inspired by contrastive learning's approach of mining supervisory signals from the data itself, this model focuses on exploring a multi-view contrastive learning mechanism to alleviate the challenges above. The main contributions of this model are as follows (shown in Figure 1):

- Incorporates random noise into attention weights to conduct contrastive learning among different

*Corresponding author
Email address: hansumuruh@xju.edu.cn

DOI reference number: 10-18293/JVLC2024-N1-050

attention weights, and introduction of a novel automatic masking mechanism for data augmentation of KG, enhancing the consistency of KG-augmented subgraphs to fortify user-item interaction graphs.

- Implementation of local-level contrastive learning between KG and user-item graphs, fostering a detailed contrastive understanding at the granular level.
- Employment of graph attention mechanisms for high-order semantic encoding of user-item-entity graphs, assigning varying weights to nodes, thereby generating embeddings for users and items that reflect their unique roles and relationships.
- Conduction of global-level contrastive learning between locally embedded nodes and the node embeddings within the user-item-entity graph, yielding more nuanced node embeddings. This process enriches representations and mitigates issues of data sparsity and noise prevalent in recommendation models.

2. Related work

2.1 Knowledge-aware recommendation

The embedding based method [5] uses Knowledge Graph Embedding (KGE) [6,7] to preprocess KG, and then incorporates the learned entity embeddings and relationship embeddings into recommendations. Collaborative Knowledge Base Embedding (CKE) [8] combines the CF module with project structure, text, and visual knowledge embedding in a unified Bayesian framework. KTUP [9] considers the incompleteness of knowledge graphs when using them for recommendation algorithms, and combines learning recommendation and knowledge graph completion. This method proposes a TUP (translation based user preference) model combined with knowledge graph learning, and utilizes multiple implicit relationships between users and items to reveal user preferences. KTUP combines TUP and TransH [10] for joint learning, enhancing project and preference modeling by transferring entity knowledge and relationships. RippleNet [11] is a classic recommendation algorithm based on knowledge graph propagation mechanism. In RippleNet, the items that users interact with are called seeds, and each seed propagates in the knowledge graph, spreading to other entities, thereby extending and expanding user interests. The embedding based method demonstrates high flexibility in utilizing KG, but the KGE algorithm focuses more on modeling strict semantic correlations (e.g. TransE [12] assumes head+relation=tail), which is more suitable for link prediction rather than recommendation.

The method of graph-based information aggregation mechanism neural networks (GNNs) [13,14] integrates multi hop neighbors into node representations to capture node features and graph structures, thus simulating long-range connectivity. KGCN [15] combines knowledge graph and graph convolutional neural network to effectively capture local neighborhood information and consider neighbor node weights for recommendation. This model samples the neighboring nodes of candidate items in the knowledge graph, and then iteratively samples the neighboring nodes for each entity, using a linear combination of neighboring node information to characterize the neighborhood information of the nodes. KGAT [16] combines the user item interaction matrix with the knowledge graph in the Collaborative Knowledge Graph (CKG) embedding layer and obtains the graph item vector representation through embedding. Then, in the attention embedding propagation layer, the item representation is enhanced by passing the propagation vector back to neighboring multi hop nodes. By calculating the relationship weights based on the attention mechanism of the knowledge graph, the node vector representation is completed after aggregating information. Finally, in the prediction layer, the user click probability is calculated and normalized through vector calculation, and recommendation is achieved. KGIN [17] models each intention as a combination of relationships in the knowledge graph to achieve better modeling capability and interpretability. In addition, this method proposes an added information aggregation scheme that recursively integrates the relationship paths of remote connections. KGIN provides interpretability for predictions by identifying influential intentions and relationship paths.

2.2 Contrastive learning

The contrastive learning approach [18,19] acquires node representations by differentiating between positive and negative examples. Initially, DGI [18] incorporated Infomax into graph representation learning, focusing on contrasting local with global node embeddings. Following this, GMI [20] proposed contrasting central nodes with their adjacent nodes, considering both node attributes and structural positions. In a similar vein, MVGRL [21] generates node and graph-level representations of neighborhoods and graph propagation from two distinct structural perspectives (including first-order graphs), and contrasts the encoded embeddings across these two views. More recently, HeCo [19] introduced learning node representations from both network pattern and meta path perspectives, conducting contrasting learning between them. KGCL [22] employs a KG augmentation scheme to mitigate noise in information aggregation. It also leverages additional supervisory signals from the KG enhancement process to guide cross-view contrastive learning, further suppressing noisy user-item interactions. However, KGCL only performs contrastive learning between the KG and user-item views, which not consider the complete semantic information in the CKE view.

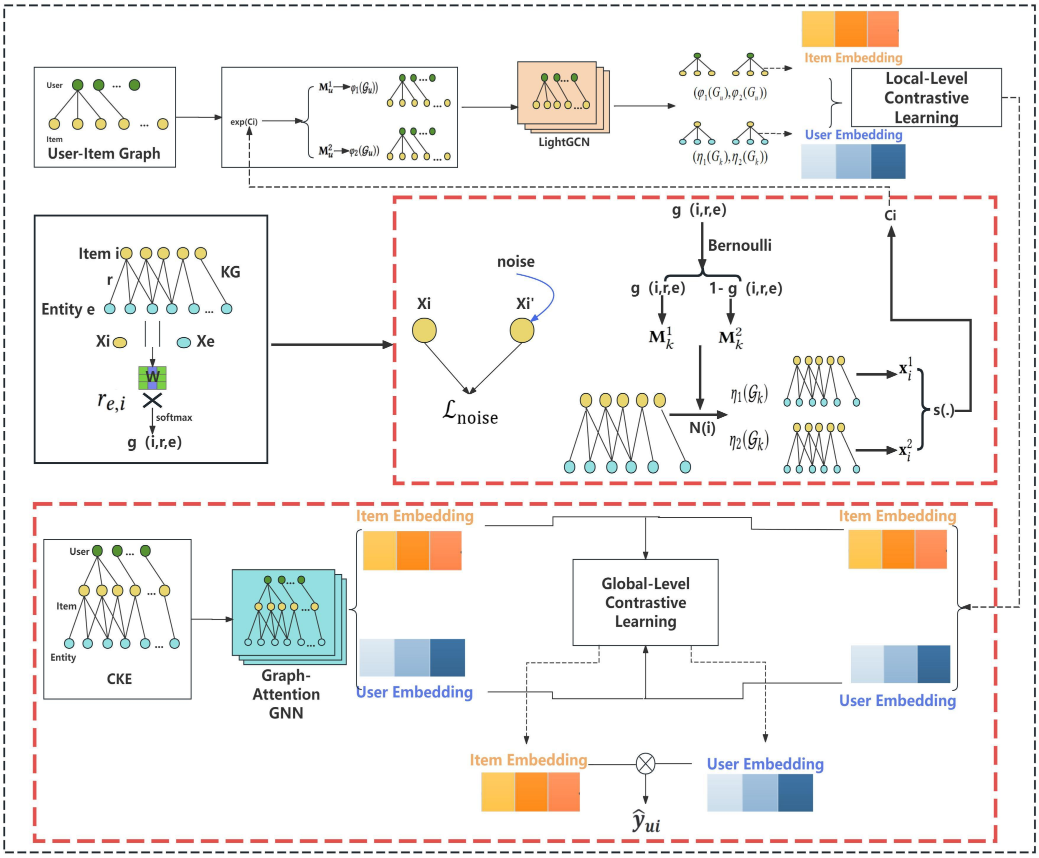

Figure 1: Main characteristics of geospatial rules.

3. Method

3.1 Preliminaries

In a recommendation scenario, we typically have historical user-item interactions such as purchases and clicks. Here, we represent this interaction data as a bipartite graph between users and items, defined as $\{(u, Y_{ui}, i) \mid u \in U, i \in I\}$, where U and I denote the sets of users and items respectively, and a connection $Y_{ui} = 1$ signifies an observed interaction between user u and item i; otherwise, $Y_{ui} = 0$.

In addition to user-item interactions, our model incorporates side information for items, comprising attributes and external knowledge that enriches item descriptions. This supplementary data involves real world entities interconnected through various relationships, effectively profiling each item. To bridge the gap between items in our primary dataset and entities within KG, we establish a mapping referred to as item-entity alignments, represented by the set $A = \{(i, r, e) | i \in I, e \in E, r \in R\}$. Each pair (i, e) in a signifies that the item i corresponds directly to an entity e within KG, thereby integrating domain-specific knowledge into our recommendation framework.

The concept of the Comprehensive Knowledge Graphs (CKG) is introduced, merging user behaviors and item knowledge into a unified graph. Each user action is depicted as a triplet (u, Y_{ui}, i), signifying an *'Interact'* relation between user u and item i when Y_{ui} = 1.Leveraging item-entity alignments, the user-item graph integrates smoothly with KG, forming a unified graph $G = \{(h, r, t)|h, t \in E', r \in R'\}$, where E' combines entities E from KG with users U $(E' = E \cup U)$, and R' expands relations R with Y_{ui} $(R' = R \cup \{Y_{ui}\})$.

3.2 CKG based graph attention network

Firstly, the TransR [23] method is used to generate the embedding representations of CKE. Consider entity h, represented by $N_h = \{(h, r, t) \mid (h, r, t) \in G\}$, which denotes the set of triples with h as the head entity. To characterize the first-order connectivity structure of entity h, this model calculates a linear combination of h's neighborhood N_h.

$$E_{N_h} = \sum_{(h,r,t)\in N_h} \pi(h,r,t)e_t \tag{1}$$

$$\pi(h,r,t) = (W_r e_t)^{\top} tanh\big((W_r e_h + e_r)\big) \tag{2}$$

Where $\pi(h,r,t)$ represents the weight parameters associated with the tail entity, and tanh is a non-linear activation function.

Following the aggregation of information for entity e_h and its neighborhood combination representation e_{Nh}, we obtain $e_h = f(e_h, e_{Nh})$, where f serves as the aggregator. We further explores higher-order connection information by gathering signals propagated from higher-hop neighbors and concatenates multi-hop vectors to achieve a final global-level representation for users and items:

$$e_h^{(l)} = f\left(e_h^{(l-1)}, e_{N_h}^{(l-1)}\right) \quad (3)$$

$$e_u^{glo} = e_u^{(0)} || \cdots || e_u^{(L)},\ e_i^{glo} = e_i^{(0)} || \cdots || e_i^{(L)} \quad (4)$$

3.3 Automatic masking mechanism

Firstly, calculate the different weights between project *i* and the entity *e* which is connected to in KG:

$$g(e, r_{e,i}, i) = \frac{exp\left(LeakyReLU\left(r_{e,i}^{\top} W[x_e || x_i]\right)\right)}{\sum_{e \in N_i} exp\left(LeakyReLU\left(r_{e,i}^{\top} W[x_e || x_i]\right)\right)} \quad (5)$$

Then, noise is added to the attention weights $g(e, r_{e,i}, i)$:

$$g'(e, r_{e,i}, i) = g(e, r_{e,i}, i) - log(-lo\, g(\epsilon)) \quad (6)$$

$$\epsilon \sim Uniform(0,1) \quad (7)$$

where ϵ is a random variable sampled from a uniform distribution. Treat the representation learned for item *i* from KG as one contrastive view, and consider the representation of item *i'* after adding random noise as another contrastive view.

$$x_i^{(l)} = e_i^{(l-1)} + \sum_{(e,r,i) \in N_i} g(e, r_{e,i}, i) x_e^{(l-1)} \quad (8)$$

$$x_{i'}^{(l)} = e_i^{(l-1)} + \sum_{(e,r,i) \in N_i} g'(e, r_{e,i}, i) x_e^{(l-1)} \quad (9)$$

The contrastive loss $\mathcal{L}_{noise}$ after adding random noise:

$$\mathcal{L}_{noise} = -log \frac{exp\left(s(x_i^{(l)}, x_{i'}^{(l)})/\tau\right)}{\sum_{i=0}^{I} exp\left(s(x_i^{(l)}, x_{i'}^{(l)})/\tau\right)} \quad (10)$$

Where s is the similarity function, and τ is the temperature parameter.

$$g(e, r_{e,i}, i) = \begin{cases} g(e, r_{e,i}, i) \in top - k\left(g(e, r_{e,i}, i)\right) \\ 0, \quad otherwise \end{cases} \quad (11)$$

Where LeakyReLU serves as the activation function, and *W* represents trainable parameters.

Secondly, unlike the random data augmentation scheme employed by the KGCL model, this model leverages the function $g(e, r_{e,i}, i)$ to generate enhancement operators M_k^1 and M_k^2 for KG triples:

$$M_k^1 = g(e, r_{e,i}, i) \quad (12)$$

$$M_k^2 = 1 - g(e, r_{e,i}, i) \quad (13)$$

Where M_k^1 and $M_k^2 \in \{0,1\}$, and finally a specific selection is made for the neighborhood N_i of item *i*:

$$\eta_1(G_k) = \left((e,r,i) \odot M_k^1\right), \eta_2(G_k) = \left((e,r,i) \odot M_k^2\right) \quad (14)$$

Wherein, the masking vectors M_k^1 and M_k^2 indicate whether specific KG triples are selected during the sampling process.

3.4 Local contrastive learning

Firstly, data augmentation is performed on the user-item view by leveraging KG to enhance consistency among subgraphs:

$$c_i = s\left(f_k\left(x_i, \eta_1(G_k)\right), f_k\left(x_i, \eta_2(G_k)\right)\right) \quad (15)$$

Where f_k denotes the aggregator function, x_i represents the embedding of items in KG.

Following this, c_i is utilized to generate two masking vectors, M_k^1 and M_k^2, which are derived from a Bernoulli distribution [24], to perform data augmentation on the user-item interaction view:

$$\varphi(G_u) = (V, M_u^1 \odot Y), \varphi(G_u) = (V, M_u^2 \odot Y) \quad (16)$$

Where V represents the set of nodes in the user-item interaction graph, and a random deletion is performed on the edge set *Y* within this interaction graph.

Following this, the nodes are encoded using the LightGCN [25] model:

$$e_u^{loc} = e_u^{(0)} + \cdots + e_u^{(L)}, e_i^{loc} = e_i^{(0)} + \cdots + e_i^{(L)} \quad (17)$$

$$\mathcal{L}_{loc} = \sum_{n \in V} -log \frac{exp(s(x_n^1, x_n^2)/\tau)}{\Sigma_{n' \in V, n' \neq n} exp(s(x_n^1, x_n^2)/\tau)} \quad (18)$$

Where s is the similarity function, and τ is the temperature parameter, (x_n^1, x_n^2) are generated from the enhanced KG and the subgraph of user-item interactions mentioned above, $\mathcal{L}_{loc}$ denotes the local contrastive loss function.

3.5 Global contrastive learning

Firstly, the node embeddings are fed into an MLP with one hidden layer:

$$z_c^{glo} = W^{(2)} \sigma\left(W^{(1)} e_c^{glo} + b^{(1)}\right) + b^{(2)} \quad (19)$$

$$z_c^{loc} = W^{(2)} \sigma\left(W^{(1)} e_c^{loc} + b^{(1)}\right) + b^{(2)} \quad (20)$$

Where $W \in R^{d \times d}$ and $b \in R^{d \times 1}$ are trainable parameters, and σ denotes the sigmoid function.

The sampling scheme for positive and negative examples is as follows: for any node in one view, the embedding of the corresponding same node learned in the other view serves as a positive example, while embeddings of all other distinct nodes are regarded as negative examples:

$$\mathcal{L}_{glo} = -log \frac{e^{s\left(z_c^{glo}, z_c^{loc}\right)/\tau}}{\underbrace{e^{s\left(z_c^{glo}, z_c^{loc}\right)/\tau}}_{\text{positive pairs}} + \underbrace{\sum_{k \neq c} e^{s\left(z_c^{glo}, z_k^{glo}\right)/\tau}}_{\text{negative pairs within the view}} + \underbrace{\sum_{k \neq c} e^{s\left(z_c^{glo}, z_k^{loc}\right)/\tau}}_{\text{negative pairs between the view}}} \quad (21)$$

To combine the recommendation task with the self-supervised task, we adopt a multi-task training strategy to optimize the entire model. Firstly, the BPR [26] loss $\mathcal{L}_{BPR}$ is constructed, therefor, the primary function after introducing the contrastive loss is as follows:

$$\mathcal{L}_{MVKGCL}=\mathcal{L}_{BPR}+\beta\big(a(\mathcal{L}_{loc}+\mathcal{L}_{noise})+(1-a)\mathcal{L}_{glo}\big)+\lambda||\Theta||_2^2 \quad (22)$$

Wherein, a and β are parameters respectively for regulating the weights of local-global contrastive loss and overall contrastive loss, while λ is the parameter that controls regularization. Details of the contrastive learning strategy are illustrated in Figure 2 below.

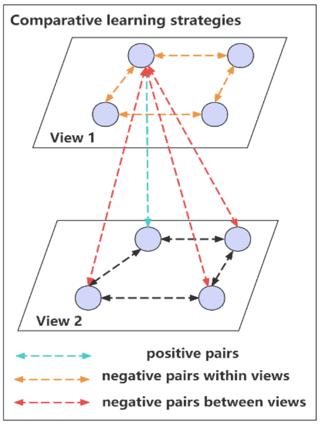

Figure 2: Contrastive learning strategies.

4. Experiments

4.1 Dataset

The experiment selects two datasets, Amazon-Book (for product recommendation) and Yelp2018 (for business venue recommendation) [27], which exhibit different levels of interaction sparsity and KG characteristics. Detailed statistical information on the datasets is provided in Table 1 below.

Table 1: Statistics of the datasets used in experiments

		Amazon-book	Yelp2018
User-Item Interactions	*#Users*	70679	45919
	#Items	24915	45538
	#Interactions	846434	1183610
KG	*#Entities*	29714	47472
	#Relations	39	42
	#Triples	686516	869603

4.2 Evaluation metrics

To evaluate the Top-N recommendation results of different models, we select two commonly used metrics for recommendation model, including Recall and Normalized Discounted Cumulative Gain (NDCG).

4.3 Evaluation protocols

The recommendation model is implemented using the Pytorch deep learning framework, with the embedding vector dimension fixed at 64 for all methods. Model optimization is carried out with a learning rate of $1e^{-3}$ and a batch size of 2048. The initial Top-k is set to 20, with both Recall and NDCG metrics considered for model evaluation. This model also conducts grid search over relevant parameters: adjusting the Top-k value within {5, 10, 20, 50, 100}, setting local contrastive loss weights to {0, 0.2, 0.4, 0.6, 0.8, 1}, and contrasting learning weights to {1, 0.1, 0.01, 0.001} respectively.

4.4 Baselines for comparison

In the experiments, we contrast the MVKGCL with several SOTA recommendation models, which can be divided into three categories:

- The first category comprises recommendation models based on traditional collaborative filtering methods: BPR [26]and GC-MC [28].
- The second category comprises recommendation models based on graph neural networks: LightGCN [25] and SGL[29].
- The third category comprises recommendation models based on knowledge graphs: CKE[8], RippleNet[11], KGCN[15], KGIN[17], CKAN[30], MVIN[31], and KGCL[22].

4.5 Performance comparison with SOTA

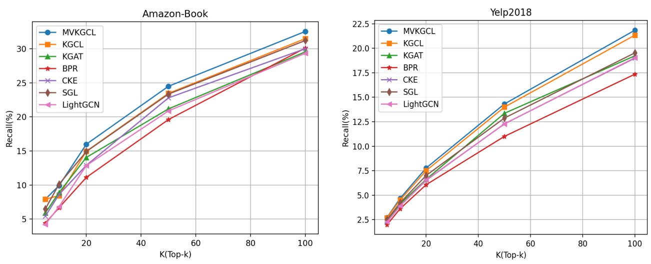

Figure 3: The result of Recall@K in top-K recommendation.

Table 2: Performance results obtained

Model	Amazonb *Recall*	*NDCG*	Yelp2018 *Recall*	*NDCG*
BPR	12.44%	0.0658	5.55%	0.0375
GC-MC	10.33%	0.0532	5.35%	0.0346
LightGCN	13.98%	0.0736	6.82%	0.0443
SGL	14.45%	0.0766	7.19%	0.0475
CKE	13.75%	0.0685	6.86%	0.0431
RippleNet	10.58%	0.0549	4.22%	0.0251
KGCN	11.11%	0.0569	5.32%	0.0338
KGIN	14.36%	0.0748	7.12%	0.0462
CKAN	13.80%	0.0726	6.89%	0.0441
MVIN	13.98%	0.0742	6.91%	0.0441
KGCL	14.96%	0.0793	7.56%	0.0493
MVKGCL	**15.86%**	**0.0846**	**7.73%**	**0.0526**

Table 2 above presents the experimental results of all models. Through observing the contrast experiments between model MVKGCL and the baseline models, the following observations are derived:

- Overall, in the experiments conducted on the Amazon-book and Yelp2018 datasets, MVKGCL demonstrates superior performance compared to other models. In terms of the Recall evaluation metric, MVKGCL achieves 15.86% and 7.73%, respectively, and for the NDCG evaluation metric, it reaches 0.0846 and 0.0526, respectively, surpassing the current SOTA model KGCL.

- In the Top-k recommendation task, our model MVKGCL surpasses the best baseline models in Recall@K metrics at multiple different K values. Among them, the joint recommendation methodologies (MVKGCL, MVIN, and KGCL) excel in recommendation effectiveness compared to embedding-based methods (CKE) and path-based methods (RippleNet). Specifically, on two datasets, MVKGCL enhances Recall by 3.5% to 4.4% and NDCG by 2.4% to 2.6% in contrast to CKE and RippleNet. The rationale behind this is that the model fully leverages the advantages of both embedding and path-based approaches, refining the depiction of entities and their relations through an iterative updating strategy, thereby compensating for the limitation of embedding methods in capturing higher-order semantic information. For more detailed outcomes, please refer to Figure 3 above.
- As a joint recommendation model, MVKGCL performs well, outperforming models such as KGIN, CKAN, and KGCL on both datasets. The main reasons for this include: firstly, the MVKGCL model, through global contrastive learning, comprehensively considers the complete structural information within the graph; furthermore, the automatic masking mechanism proposed in this paper adequately considers the varying attention weights between items and entities, selectively enhancing the data of KG.

4.6 Ablation study of MVKGCL

The experiment investigates the model's performance from the perspectives of KG and contrastive learning. Ablation experiments can verify the functions of different components of the model. MVKGCLw/o glo denotes the model variant without global contrastive learning, a component primarily utilizing an automatic masking mechanism to augment KG data. Meanwhile, MVKGCLw/o mask signifies the model version sans the automatic masking mechanism, which employs GAT for embedding learning on CKE graphs; subsequently, it leverages global contrastive learning to derive node embeddings enriched with semantic and structural information at the global level. Each component in the MVKGCL model contributes positively, with the complete MVKGCL model outperforming both MVKGCLw/o glo and MVKGCLw/o mask across evaluation metrics on two datasets. Furthermore, MVKGCLw/o glo consistently surpasses MVKGCLw/o mask in all metrics, highlighting the efficacy of employing attention weights to generate mask vectors for data augmentation of KG. The experimental results are shown in Table 3 below.

Table 3: Impact study of MVKGCL model variants

Model	Amazon-book		Yelp2018	
	Recall	*NDCG*	*Recall*	*NDCG*
MVKGCL	15.86%	0.0846	7.73%	0.0526
MVKGCLw/o glo	15.66%	0.0829	7.63%	0.0511
MVKGCLw/o mask	15.60%	0.0820	7.60%	0.0502

4.7 Impact of local-level contrastive loss weight

Investigating the impact of local versus global contrast weights on model metrics. Specifically, to study the effect of the weight parameter a, the model varies a's value within the set {0, 0.2, 0.4, 0.6, 0.8, 1.0}, leading to the following observations: (1) For the Amazon-Book dataset, the model achieves its best performance when a=0.4; for the Yelp2018 dataset, the optimal performance is reached when a=0.2, indicating that at these points, a balance between local and global contrastive losses is achieved; (2) In the case of the Amazon-Book dataset, the worst performance typically occurs when a=0, highlighting the significance of the automatic masking mechanism. For the Yelp2018 dataset, poor performance is observed when the parameter a is either 0 or 1, suggesting that both levels of contrastive loss play a crucial role in the model's functioning. See Figure 4 below for further details.

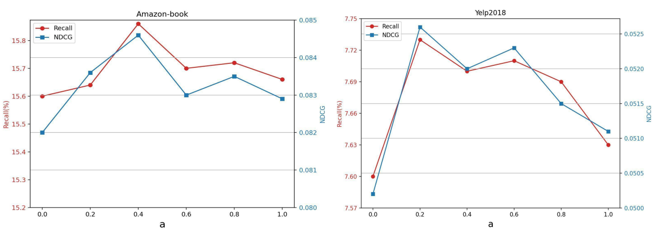

Figure 4: Impact of local-level contrastive loss weight.

4.8 Impact of contrastive loss weight

By adjusting the weights of contrastive learning, we explore the role contrastive learning plays in the model to uncover the significance of contrastive loss during multi-task training. Specifically, we vary the parameter β within the set {1, 0.1, 0.01, 0.001}, observing performance metrics across different datasets. The experimental results indicate that the model performs best when the parameter β is set to 0.1. The primary reason for this improvement is the adjustment of the contrastive loss to a level comparable with the recommendation task loss, thereby enhancing the model's performance. Details are provided in Table 4 below.

Table 4: Impact of contrastive loss

	Amazon-book		Yelp2018	
	Recall	*NDCG*	*Recall*	*NDCG*
β=1	15.63%	0.0823	7.70%	0.0513
β=0.1	15.86%	0.0846	7.73%	0.0526
β=0.01	15.58%	0.0818	7.69%	0.0511
β=0.001	15.40%	0.0806	7.67%	0.0510

4.9 Vector embedding representation

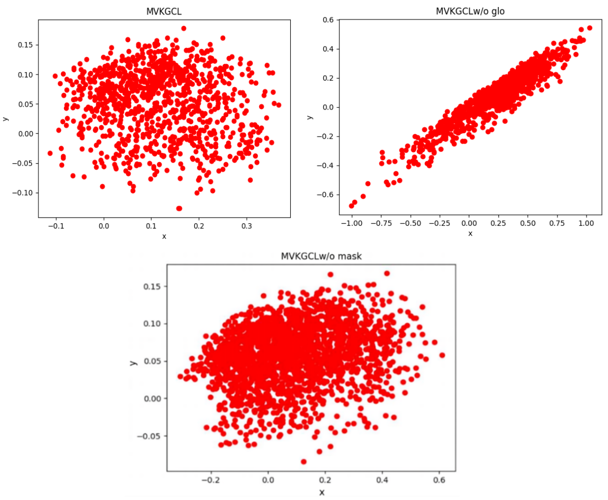

Figure 5: Project Embedding Representation in Amazon-Book.

To evaluate whether the contrast mechanism affects the performance of representation learning, this paper employs SVD decomposition to embed items into a two-dimensional space. As shown in Figure 5, this work contrasts the visualization results of MVKGCL, MVKGCLw/o glo, and MVKGCLw/o mask on the Amazon-book dataset. The following observations can be drawn from the figure below:

- The item node embeddings generated by MVKGCLw/o mask are mixed to some extent, while those produced by MVKGCLw/o glo fall into a narrow cone shape. In contrast, the node embeddings generated by the MVKGCL model exhibit a more diverse distribution: specifically, they are distributed more evenly and sparsely, thereby capable of representing different node feature information. This indicates that the MVKGCL model has superior capabilities in representation learning and mitigating representation degradation.
- By contrasting MCCLK and its variants, it is observed that removing the auto-mask or the global contrastive learning component makes the embedding representations less distinguishable. This evidence supports that the MVKGCL model enhances the effectiveness and robustness of representation learning.

5. Conclusion and future work

The work proposes a recommendation model MVKGCL based on KG and contrastive learning: firstly, an automatic masking mechanism is introduced to augment the data in KG; secondly, by employing graph attention neural networks, the complete structural information within CKG is mined. The node embeddings learned from this process are then globally contrasted with node embeddings obtained through local contrastive learning, fully exploiting the structural and semantic information within KG. Experiments constructed demonstrate that MVKGCL outperforms other existing models in terms of performance.

In future work, to address the issue of inadequate exploitation of structural views by the model, a new paradigm of graph attention neural networks will be considered for feature optimization of structural views. This enhancement aims to further elevate the model's performance.

References

[1] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In WWW. 689–698.

[2] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural collaborative filtering vs. matrix factorization revisited. In Recsys. 240–248.

[3] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In KDD. 950–958.

[4] Guanying Wang, Wen Zhang, Ruoxu Wang, Yalin Zhou, Xi Chen, Wei Zhang, Hai Zhu, and Huajun Chen. 2018.Label-free distant supervision for relation extraction via knowledge graph embedding. In EMNLP. 2246–2255.

[5] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019.Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In WWW. 151–161.

[6] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In AAAI.

[7] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In AAAI.

[8] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016.Collaborative knowledge base embedding for recommender systems. In SIGKDD.353–362.

[9] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019.Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In WWW. 151–161.

[10] Wang Z, Zhang JW, Feng JL, et al. Knowledge graph embedding by translating on hyperplanes. Proceedings of the 28th AAAI Conference on Artificial Intelligence. Quebec City: AAAI, 2014. 1112–1119.

[11] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie,and Minyi Guo. 2018. Ripplenet:

Propagating user preferences on the knowledge graph for recommender systems. In CIKM. 417–426.

[12] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Ok sana Yakhnenko. 2013.Translating embeddings for modeling multi-relational data. In Neural Information Processing Systems (NIPS). 1–9.

[13] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018.Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In SIGKDD. 1531–1540.

[14] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. 2018.Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering (2018), 357–370.

[15] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019.Knowledge graph convolutional networks for recommender systems. In WWW. 3307–3313.

[16] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019.Kgat:Knowledge graph attention network for recommendation. In SIGKDD. 950–958.

[17] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning Intents behind Interactions with Knowledge Graph for Recommendation. In WWW. 878–887.

[18] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. ICLR (Poster) (2019), 4.

[19] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. Self-supervised Hetero geneous Graph Neural Network with Co-contrastive Learning. arXiv preprint arXiv:2105.09111 (2021).

[20] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph representation learning via graphical mutual information maximization. In WWW. 259–270.

[21] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In ICML. PMLR, 4116–4126.

[22] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge Graph Contrastive Learning for Recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computing Machinery, New York, NY, USA, 1434–1443.

[23] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In AAAI.

[24] Albert W Marshall and Ingram Olkin. 1985. A family of bivariate distributions generated by the bivariate Bernoulli distribution. J. Amer. Statist. Assoc. 80, 390(1985), 332–338.

[25] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In SIGIR. 639–648.

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv (2012).

[27] Ding Zou, Sen Zhao, Wei Wei, Xian-ling Mao, Ruixuan Li, Dangyang Chen, Rui Fang, and Yuanyuan Fu. 2023. Towards Hierarchical Intent Disentanglement for Bundle Recommendation. IEEE Transactions on Knowledge and Data Engineering(2023).

[28] Rianne van den Berg, Thomas N Kipf, and Max Welling.2017. Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263.

[29] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In SIGIR.726–735.

[30] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. 2020. CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems. In SIGIR. 219–228.

[31] YChang-You Tai, Meng-Ru Wu, Yun-Wei Chu, Shao-Yu Chu, and Lun-Wei Ku. 2020. MVIN: Learning Multiview Items for Recommendation. In SIGIR. 99–108.