Dual-View Self-supervised Co-training for Knowledge Graph Recommendation

Ruoyi Zhang[1], Huifang Ma[1,2](✉), Qingfeng Li[1], Yike Wang[1], and Zhixin Li[2]

[1] College of Computer Science and Engineering, Northwest Normal University, Lanzhou, Gansu, China
mahuifang@yeah.net

[2] Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin, Guangxi, China

Abstract. Knowledge Graph Recommendation (KGR), which aims to incorporate Knowledge Graphs (KGs) as auxiliary information into recommender systems and effectively improve model performance, has attracted considerable interest. Currently, KGR community has focused on designing Graph Neural Networks (GNNs)-based end-to-end KGR models. Unfortunately, existing GNNs-based KGR models are focused on extracting high-order attributes (knowledge) but suffer from restrictions in several vital aspects, such as 1) neglect of finer-grained feature interaction information via GNNs and 2) lack of adequate supervised signals, leading to undesirable performance. To tackle these gaps, we propose a novel **D**ual-view **S**elf-supervised **C**o-training for **K**nowledge **G**raph Recommendation (DSCKG). We consider two different views, covering user-item collaborative view and KGs structural view. Precisely, for the collaborative view, we first extract high-order collaborative user/item representations with GNNs. Next, we impose a discrepancy regularization term to augment the self-discrimination of the user/item representations. As for the structural view, we initially utilize GNNs to extract high-order features. Next, we utilize novel Dual-core Convolutional Neural Networks to extract bit- and vector-level finer-grained feature interaction signals. DSCKG hence performs a high-quality self-supervised co-training paradigm across dual views, improving the node representation learning capability. Experimental results demonstrate DSCKG achieves remarkable improvements over SOTA methods.

Keywords: Knowledge Graph Recommendation · Convolutional Neural Networks · Graph Neural Networks · Self-supervised Learning

1 Introduction

Nowadays, Recommender systems aim to alleviate information overload, and are pivotal in fields such as social networks [17]. The classical recommendation approach: Collaborative Filtering (CF), which aims at modeling user preferences based on user-item similarities of historical interactions. However, CF-based recommendation models often fall into the dilemma of data sparsity and cold-start issues. To alleviate the above dilemma, researchers have suggested incorporating side information into recommendation model. Knowledge Graphs (KGs), which consist of real-world objective facts and fruitful entities, are significantly helpful for improving recommendation performance. Hence, Knowledge Graph Recommendation (a.k.a., KGR) has recently attracted considerable attention.

KGR's development lines can be divided into three phases: 1) Embedding-based KGR approaches, which mainly utilize knowledge graph embedding (e.g., TransR) techniques to learn high-quality entity-relationship representations. Unfortunately, the method is unable to extract long-range connections, leading to unsatisfactory recommendation performance. 2) Path-based KGR methods, which mainly exploit meta-path techniques [3] to extract long-range connections [20]. However, this type of methods require expert knowledge with manually defined meta-paths, which is not effective in practice. Meanwhile, the widespread utilization of 3) Graph Neural Networks (GNNs) and Graph Attention networks (GATs) [8] techniques, which can adaptively extract high-order knowledge (attribute information), leads to State-Of-The-Art (SOTA) for downstream recommendation tasks.

Primary Motivation. Despite the promising performance of current GNNs-based KGR methods, the following two aspects are still under-explored:

- **Existing GNNs-based KGR methods fail to extract feature interaction signals**. Current GNNs-based KGR methods for integrating high-order features commonly select mechanistic aggregation strategies (e.g., max-pooling or summation) without feature interaction signals, which is insufficient. On the one hand, selecting conventional and naive high-order feature aggregation strategies could easily lead to over-smoothing and representation degradation issues. On the other hand, potentially valuable finer-grained feature interaction signal is underutilized, which would greatly hinder the representation capability of the model. Intuitively, designing a finer-grained feature interaction mode among high-order features improves model representation capability and recommendation performance.
- **Supervision interaction data scarcity**. Graph-based methods are more susceptible to data sparsity dilemmas. Since established in a supervised mode, existing GNNs-based KGR methods rely on observed user-item inter-actions as supervised signals to perform graph representation learning in KGs. However, practical training often implies a lack of training labels because of the sparsity problem inherent in recommender systems. This issue causes the model failing to learn reliable representations, thus limiting the performance gains of the KGR model.

Considering the limitations of these two lines, we believe that it is essential to properly model finer-grained feature interaction and explore high-quality self-supervised signals in the KGR task.

Towards this end, we a novel **D**ual-view **S**elf-supervised **C**o-training for **K**nowledge **G**raph Recommendation (DSCKG). Specifically, upon the user-item collaborative view, we employ a LightGCN [2] to generate prototype user/item representations. Besides, we impose a novel discrepancy regularization strategy to encourage user/item representations independence from each other. Upon the KGs structural view, we first build the GNNs-based knowledge-aware attention module as the backbone networks to obtain the user/item embeddings. Next, we execute the finer-grained feature interaction model for the item-side. Specifically, we innovatively employ the Dual-core Convolutional Neural Networks (DuCNNs) to extract bit-level (vertical convolution) and vector-level (horizontal convolution) finer-grained feature interaction signals. Immediately after, we rely on two views, which collaboratively utilize different view information to generate extra supervised signals to guide the multi-view self-supervised co-training paradigm. Finally, we jointly employ the cross-entropy primary loss and the rest of the regularization terms to optimize the DSCKG. Overall, we summarize the three-fold contributions of the proposed DSCKG as follows:

- **Comprehensive aspects.** We point out the issues of neglecting finer-grained feature interaction modelling and the lack of supervised signals in KGR tasks, and we advocate the employment of finer-grained feature interactions and self-supervised co-training strategies to mitigate the above issues for better node representation learning.
- **Novel Approach.** We propose a new model, DSCKG, which designs a novel self-supervised co-training mechanism for KGR. Precisely, there are three major novelties: 1) We design a novel discrepancy regularization strategy to encourage the generated user/item representation to be self-identifying on the collaborative view. 2) DSCKG designs a new CNNs-based finer-gained feature interaction pattern for KGs structural view to enhance item-side representations. 3) We propose a novel self-supervised co-training mode to identify high-quality positive/negative samples better.
- **Multi-faceted Experiments.** We perform empirical experiments on three real-world KGs-based recommendation datasets whose results demonstrate the superiority of DSCKG over compelling SOTA baselines.

2 Related Work

Next, we review the most relevant current work related to the proposed approach: 1) GNNs-based KGR methods, and 2) Self-supervised modes in recommendations.

GNNs-Based Recommendation Methods. The GNNs-based KGR approaches are established on the high-order neighbor information aggregation mechanism on graphs, which can iteratively carry out the propagation mechanism to integrate multi-hop neighbor information into node representations. KGAT [14] integrates user-item-entity into a unified heterogeneous Collaborative Knowledge Graph (CKG) and recursively aggregates CKG via **G**raph **A**ttention Ne**T**works (GATs [8]). KGCN [11] and KGNN-LS [10] primarily recursively integrate item-side high-order neighbor information via Graph Convolutional Networks (GCNs) to obtain item-side high-order embeddings. KGIN [15] utilizes intent-aware techniques to disentangle the intent behind user-item interactions, performing recursive aggregation operations on KGs by leveraging relation-aware mechanisms. CKAN [16] employs a heterogeneous propagation mechanism that can enrich user- and item-side representations by capturing CF signals as well as knowledge-aware signals through an attention mechanism. Regrettably, mainstream GNNs-based methods have three significant drawbacks: 1) Failure to model finer-grained feature interactions; 2) Inevitable over-smoothing phenomenon; 3) Serious issue of representation degradation.

Self-supervised Modes in Recommendations. Self-supervised methods aim to build correct positive/negative sample pairs to direct better node/graph representation learning. The recommender systems community has inherited this technological trend and applies self-supervised learning modes to various recommendation tasks [4]. One line of current approaches is dedicated to graph augmentation techniques. To be precise, new views(a.k.a., views) are generated based on perturbations of the original data, e.g., SGL [18] uses three graph augmentation methods (i.e., node/edge dropout, and random walk). Another line of current approaches is aimed at constructing different views, e.g., COTREC [19] constructs item and session views for high-quality self-supervised learning. CrossCBR [5] constructs complementary views based on two data sources for bundle recommendation. Unfortunately, current work has not fully explored the vast potential of self-supervised learning and KGs-based recommendation.

3 Preliminaries

User-Item Interaction Data. In a typical recommendation case, we have a user set of M users $U = \{u_1, \ldots, u_M\}$ and a item set of N items $V = \{v_1, \ldots, v_N\}$. The interaction matrix $\mathbf{Y} \in \{0, 1\}^{|M| \times |N|}$ is determined by user-item (u, v) implicit feedback (i.e., $y_{u,v} = 1$ indicates that an interaction between u and v is observed, otherwise $y_{u,v} = 0$).

Knowledge Graph. Let Knowledge Graph $G = (\mathcal{E}, \mathcal{R})$, which is an un-directed graph composed of knowledge triples (h, r, t). Where $h, t \in \mathcal{E}$, $r \in \mathcal{R}$ are on behalf of head and tail-entity, as well as relation of a knowledge triple correspondingly. $\mathcal{E}$ and $\mathcal{R}$ denotes the entities sets and relations in KGs. Besides, we define an item-entity alignment set $\mathcal{A} = \{(v, e) \mid v \in V, e \in \mathcal{E}\}$ that is denoted to uncover the alignment operations of items in both the user-item interaction matrix and KGs. Ultimately, we aim to learn a match function $\tilde{y}_{uv} = \mathcal{F}(u, v|\Theta, \mathbf{Y}, G)$, where $\tilde{y}_{uv}$ defines the possibility that a user u would adopt a item v, and Θ is the model parameters set.

Task Description

- Input: Knowledge Graph G, U-I interaction matrix $\mathbf{Y}$, and the model parameters set Θ.
- Output: The probability $\tilde{y}_{uv}$ that the user u interacts with the item v.

4 DSCKG Model Methodology

Overview. The framework of DSCKG is shown in Fig. 1. It consists of three key modules. 1) Collaborative View Encoder: It relies on historical user-item

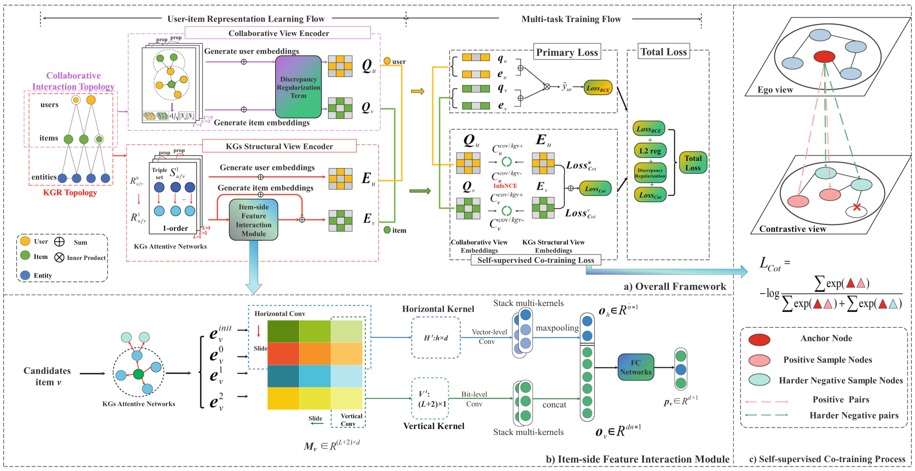


Fig. 1. Illustration of the proposed DSCKG framework. Best viewed in color.

interaction data to build user/item collaborative view prototype embeddings. And imposes a discrepancy regularization term on the generated embeddings. 2) Structural view encoder: It uses knowledge-aware attention networks to generate user/item prototype embeddings. And CNN is used to extract finer-grained item-side feature interaction embeddings. 3) Multi-task training module: Joint primary loss and the rest of optimization terms to guide the training and optimization of the whole model.

4.1 Collaborative View Encoder

We consider utilizing the GNNs-based method by performing iterative propagation on the user-item interaction graph to generate a high-order representation of the user/item from collaborative view. Inspired by the precious Collaborative Filtering (CF) work, we utilize LightGCN to model the long-range connectivity from user-item interactions. LightGCN [2] dispenses with complex message passing as well as feature fusion mechanisms, and eliminates the need for feature transformation and nonlinear activation functions and have high computational efficiency. In the $l'+1$-th layer, the aggregation proceeding can be formulated as follows:

$$\boldsymbol{q}_u^{(l'+1)} = \sum_{v\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|\,|\mathcal{N}_v|}} \boldsymbol{q}_v^{(l')}, \boldsymbol{q}_v^{(l'+1)} = \sum_{u\in\mathcal{N}_v} \frac{1}{\sqrt{|\mathcal{N}_u|\,|\mathcal{N}_v|}} \boldsymbol{q}_u^{(l')} \qquad (1)$$

Here $l' \in \{0, 1, 2, \ldots, L'\}$. After L'-order propagation, we adopt the *sum* aggregator as the high-order integration function to incorporate all layer embeddings to yield the final embedding $\boldsymbol{q}_u$ and $\boldsymbol{q}_v$ as such:

$$\boldsymbol{q}_u = \sum_{l'=0}^{L'} \boldsymbol{q}_u^{(l')}, \boldsymbol{q}_v = \sum_{l'=0}^{L'} \boldsymbol{q}_v^{(l')} \tag{2}$$

Discrepancy Regularization Term. The user/item prototype representations generated by LightGCN should contain diversity and preference information. Due to the plague of data sparsity and long-tail issues, the extracted collaborative interaction signals will have some degree of information overlap and redundancy, thus negatively affecting the model. Besides, with the increase of LightGCN depths, node embeddings are extremely vulnerable to local similarity. We employ an embedding optimization strategy to encourage representations of self-discrimination and independence from each other [15]. Precisely, we introduce a discrepancy regularization module to guide the prototype user/item representation learning of diversity.

Inspired by metric learning techniques, we minimize the mutual information between two different embeddings to quantify their difference. Such an idea can be considered an ego-view self-supervised learning technique. More specifically, taking the user-side embedding optimization process as an instance, the discrepancy regularization term modelling is:

$$\mathcal{L}_{Dis}^{u} = \sum_{u \in U} -\log \frac{e^{s(\boldsymbol{q}_u, \boldsymbol{q}_u)/\tau'}}{\sum_{u' \in U} e^{s(\boldsymbol{q}_u, \boldsymbol{q}_{u'})/\tau'}} \tag{3}$$

Here $s(\cdot)$ is shown as the cosine similarity function between any two embeddings, and τ' is denoted the temperature coefficient. The item-side $\mathcal{L}_{Dis}^{v}$ and the user-side $\mathcal{L}_{Dis}^{u}$ have the same construction flow. Therefore, the overall discrepancy regularization loss is:

$$\mathcal{L}_{Dis} = \mathcal{L}_{Dis}^{u} + \mathcal{L}_{Dis}^{v} \tag{4}$$

We optimize discrepancy regularization loss to encourage divergence between different user/item embeddings and allow these embeddings to contain richer semantic information for the aim of embedding optimization.

4.2 KGs Structural View Encoder

Backbone Networks. The KGs structural view encoder relies on a knowledge-aware attention mechanism to extract high-order neighbor feature information, which follows the prior KGR approach [9,16]. The 0 and l-order receptive field R are denoted as:

$$\begin{cases} R_u^0 = \{e \mid v \in \{v \mid y_{uv} = 1\} \wedge (v, e) \in \mathcal{A}\} \\ R_v^0 = \{e \mid v^* \in \{v^* \mid \exists u \in U, y_{uv^*} = 1 \wedge y_{uv} = 1\} \wedge (v^*, e) \in \mathcal{A}\} \end{cases} \tag{5}$$

$$R^l_{u/v} = \left\{ t \mid (h, r, t) \in G \wedge h \in R^{l-1}_{u/v} \right\} \tag{6}$$

where $l \in \{1, 2, \dots, L\}$. Next, we elaborately describe the user u and item v 1-order representation learning process. We also need to define the l-th knowledge triples set S. The triplet set size directly affects the size of related high-order KGs entities.

$$S^l_{u/v} = \left\{ (h, r, t) \mid (h, r, t) \in G \wedge h \in R^{l-1}_{u/v} \right\}. \tag{7}$$

We demonstrate the process of computing 1-order embeddings via the knowledge-aware attention mechanism as such:

$$\alpha_{att} = Softmax \left(\text{MLP} \left[\boldsymbol{e}^h_i \| \boldsymbol{e}^r_i \right] \right) \tag{8}$$

Here $i \in \{1, 2, \dots, \left| S^1_{u/v} \right| \}$, and MLP$(\cdot)$ denotes a multi-layer perceptron (MLP) and $||$ denotes the vector *concat* operation. Then we can obtain the user and item embedding $\boldsymbol{e}^1_u$ and $\boldsymbol{e}^1_v$ after 1-layer propagation as follows:

$$\boldsymbol{e}^1_{u/v} = \sum_{i=1}^{\left| S^1_{u/v} \right|} \alpha_{att} \cdot \boldsymbol{e}^t_i \tag{9}$$

Besides, we introduce the user/item 0-order embeddings $\boldsymbol{e}^0_u, \boldsymbol{e}^0_v$ and the item v itself initial embedding $\boldsymbol{e}^{init}_v$ as follows:

$$\boldsymbol{e}^0_{u/v} = \frac{1}{\left| R^0_{u/v} \right|} \sum_{e \in R^0_{u/v}} \boldsymbol{e}, \boldsymbol{e}^{init}_v = \frac{1}{|\{e \mid (v, e) \in \mathcal{A}\}|} \sum_{\{e|(v,e) \in \mathcal{A}\}} \boldsymbol{e} \tag{10}$$

The high-order user/item embeddings can be generalized from the 1-order embedding computation procedure described above to derive the user/item high-order embedding sets naturally: $\left\{ e^0_u, e^1_u, \dots, e^L_u \right\}, \left\{ e^{init}_v, e^0_v, e^1_v, \dots, e^L_v \right\}$. And we integrate high-order embeddings by *sum* aggregator. Thus, we acquire user u and unfinished item v integration embeddings: $\boldsymbol{e}_u$ and $\tilde{\boldsymbol{e}}_v$.

DuCNNs. Current GNNs-based KGR approaches acquire the user/item ultimate embedding (e.g. KGAT [14], CKAN [16], and KGIN [15]) only by mechanically aggregating high-order feature representations while ignoring finer-grained feature interactions modelling [22], which is insufficient. Finer-grained feature interaction modelling can further explore the potential feature co-occurrence signals in the KGR task, which can directly improve the recommendation performance. Thus, we firmly believe finer-grained feature interaction are also urgently needed and deficient for the KGR task. We creatively generalize CNNs to the KGR task and utilize them as high-order feature aggregators for item-side 2-D multi-order embedding matrix $\boldsymbol{M}_v \in \mathbb{R}^{(2+L) \times d}$. We innovatively apply two different convolution kernels (i.e., **Horizontal** and **Vertical convolutional kernel**) for capturing two diverse high-order feature interaction signals.

Exactly, we denote vertical convolutional kernel $\boldsymbol{V}^t \in \mathbb{R}^{(2+L) \times 1}$ slides column by column on the matrix $\boldsymbol{M}_v$ to extract dimension-specific finer-grained feature interaction signals, referred the *bit-level feature interaction pattern*. Analogously, we denote the horizontal convolutional kernel $\boldsymbol{H}^t \in \mathbb{R}^{h \times d}$ slides matrix

$\boldsymbol{M}_v$ by rows to extract feature interaction signals between neighboring features, referred the *vector-level feature interaction mode*. In addition, to obtain richer feature interaction information, we stack multiple [8] convolutional kernels for both classes of convolutional patterns. The following is the specific workflow of Dual-core Convolutional Neural Networks (DuCNNs).

Vector-Level Feature Interaction Mode. A horizontal convolutional kernel $\boldsymbol{H}^t \in \mathbb{R}^{h \times d}$ is applied to extract the adjacent-order feature interaction signals, and the detailed process is shown in the subgraph-b upper part of Fig. 1. It is necessary to employ multiple convolutional kernels when we employ the multi-head mechanism, with the aim of capturing richer feature interaction information, as described above. Hence, $h \in \{1, 2, \ldots, (2+L)\}$ is referred to the horizontal convolution kernel height and $t \in \{1, 2, \ldots, \tilde{n}\}$. The following equation shows the computation of the i-th convolutional value:

$$\tilde{\boldsymbol{c}}_i^t = \phi_c \left(\boldsymbol{M}_v[i : i+h-1, :] \odot \boldsymbol{H}^t \right) \tag{11}$$

where $\phi_c(\cdot)$ is denoted as the activation function (ReLU) of the convolutional layer, and $\odot$ is denoted as the inner product symbol. The horizontal convolutional output result (vector) $\tilde{\boldsymbol{c}}^t \in \mathbb{R}^{(2+L)-h+1}$ is expressed as:

$$\tilde{\boldsymbol{c}}^t = \left[\tilde{c}_1^t, \ldots, \tilde{c}_i^t, \ldots, \tilde{c}_{((2+L)-h+1)}^t \right] \tag{12}$$

We require extracting the highest significant feature interaction signals in the vector-level feature interaction pattern and filtering out the noise signals unrelated to the feature interaction as much as possible. Hence, we implement the *maxpooling* operation and the final horizontal convolutional vector $\boldsymbol{o}_h \in \mathbb{R}^{\tilde{n}}$ is:

$$\boldsymbol{o}_h = \left[\; maxpooling \left(\tilde{\boldsymbol{c}}^1 \right), \ldots, \; maxpooling \left(\tilde{\boldsymbol{c}}^{\tilde{n}} \right) \right] \tag{13}$$

Bit-Level Feature Interaction Mode. The vertical convolutional kernel $\boldsymbol{V}^t \in \mathbb{R}^{(2+L) \times 1}$ is applied to extract dimension-specific significant feature interactions information, and the detailed process is shown in the subgraph-b lower part of Fig. 1. Then the kernel $\boldsymbol{V}^t$ overlays the 2-D embedding matrix $\boldsymbol{M}_v$ and shift along the dimensional direction. Likewise, the following equation shows the computation of the i-th convolutional value:

$$\boldsymbol{c}_i^t = \phi_c \left(\boldsymbol{M}_v[:, i] \odot \boldsymbol{V}^t \right) \tag{14}$$

where $\boldsymbol{V}^t \in \mathbb{R}^{(2+L) \times 1}$. The vertical convolutional output result (vector) $\boldsymbol{c}^t \in \mathbb{R}^d$ is:

$$\boldsymbol{c}^t = \left[c_1^t, \ldots, c_i^t, \ldots, c_d^t \right] \tag{15}$$

Here d is denoted as the fixed embedding dimension. We concatenate n kernels in sequence. The *maxpooling* operation is discarded because the feature information of each dimension needs to be retained to the maximum extent. The final vertical convolutional $\boldsymbol{o}_v \in \mathbb{R}^{dn}$is given by:

$$\boldsymbol{o}_v = \left[\boldsymbol{c}^1 || \boldsymbol{c}^2 || \ldots || \boldsymbol{c}^n \right] \tag{16}$$

where $||$ is shown as the vector connection operator.

Fully Connected (FC) Networks. We concatenate the above two convolutional vectors, fed into FC networks to extract global interactive features [7], and output the item feature-enhanced convolutional embedding $\boldsymbol{p}_v$ as such:

$$\boldsymbol{p}_v = \sigma\left(\mathbf{W} \cdot [\boldsymbol{o}_v \| \boldsymbol{o}_h]\right) \tag{17}$$

where $\mathbf{W} \in \mathbb{R}^{d \times (\tilde{n}+dn)}$ is the transformation matrix and the convolutional embedding $\boldsymbol{p}_v \in \mathbb{R}^d$. Then the final item-side representation: $\boldsymbol{e}_v = \tilde{\boldsymbol{e}}_v + \boldsymbol{p}_v$.

4.3 Dual-View Self-supervised Co-training Mode

We present our novel Dual-view self-supervised co-training mode in terms of the quality of the selected harder negative sample [24]. We consider that high-quality negative samples have the following quality: Having a sufficient and significant contribution of self-supervised signal, called **harder negative samples**. Moreover, we firmly believe that an appropriately large positive sample size can also help implement the self-supervised learning strategy. We deploy a novel scheme to search for positive/negative samples of **anchor embedding** $\boldsymbol{e}_v$ as follows:

$$\boldsymbol{\phi}_v^{cov} = Softmax\left(\boldsymbol{value}_v^{cov}\right), \boldsymbol{value}_v^{cov} = \boldsymbol{Q}_v \cdot \boldsymbol{q}_v \tag{18}$$

Here $\boldsymbol{Q}_v \in \mathbb{R}^{N \times d}$ are the matrix of users samples in the collaborative view, $\boldsymbol{q}_v \in \mathbb{R}^{d \times 1}$ is the anchor embedding (target item v) in the collaborative view, $\boldsymbol{\phi}_v^{cov} \in \mathbb{R}^{N \times 1}$ is shown as the probability of the target item v with other items in collaborative view.

With the calculated similarity scores, we select the top-k items with the highest confidence level as a positive sample. Formally, the positive sample set $C_v^{cov^+}$ is selected as follows:

$$C_v^{cov^+} = \text{top}-k\left(\boldsymbol{\phi}_v^{cov}\right) \tag{19}$$

One of the simplest methods is to classify all items other than the top-k items of the similarity as negative samples. However, the contribution of such an approach to self-supervised learning is limited.

Therefore, we randomly select k negative samples from the items ranked in top 10% in $\boldsymbol{\phi}_v^{cov}$ excluding the positive samples to construct harder negative samples set $C_v^{cov^-}$, these items can be regarded as harder negative samples that can supply sufficient self-supervised signals. Symmetrically, we adopt the same method to choose suitable positive/negative samples for anchor embedding $\boldsymbol{q}_v$ in the KGs structural view, and to construct $C_v^{kgv^-}/C_v^{kgv^+}$. Consequently, for the generated pseudo-labels, we optimize the node representation learning for dual-views by self-supervised learning mode. Since the calculation process $\mathcal{L}_v^{cov}/\mathcal{L}_v^{kgv}$ are symmetric, we only show the calculation process of $\mathcal{L}_v^{kgv}$:

$$\mathcal{L}_v^{kgv} = -\log \frac{\sum_{i \in C_v^{cov^+}} e^{s(\boldsymbol{e}_v, \boldsymbol{q}_i)/\tau}}{\sum_{j \in \left\{C_v^{cov^+} \cup C_v^{cov^-}\right\}} e^{s(\boldsymbol{e}_v, \boldsymbol{q}_j)/\tau}} \tag{20}$$

We follow the InfoNCE [6] paradigm. Where τ and $s(\cdot)$ are consistent with the previous definitions (*c.f.*, Eq. (3)). We perform the same operation on the user-side. Hence, the overall formalization of $\mathcal{L}_{Cot}$ as such:

$$\mathcal{L}_{Cot} = \sum_{v \in V} \left[\mathcal{L}_v^{kgv} + \mathcal{L}_v^{cov}\right] + \sum_{u \in U} \left[\mathcal{L}_u^{kgv} + \mathcal{L}_u^{cov}\right] \tag{21}$$

4.4 Model Prediction and Optimization

We integrate the representations of the two views and calculate the user-item matching scores by the inner product:

$$\tilde{y}_{uv} = \sigma\left(\left[\boldsymbol{e}_u + \boldsymbol{q}_u\right]^\top \left[\boldsymbol{e}_v + \boldsymbol{q}_v\right]\right) \tag{22}$$

where $\sigma(\cdot)$ is the *sigmoid* function. Next, We adopt the Binary Cross-Entropy (BCE) loss as primary loss, and the overall loss function is shown as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{BCE} + \lambda_1 \cdot \mathcal{L}_{Cot} + \lambda_2 \cdot \mathcal{L}_{Dis} + \lambda_3 \cdot \|\Theta\|_2^2 \tag{23}$$

Here λ_1 is the hyper-parameter to determine the self-supervised loss ratio. λ_2 is the hyper-parameter to determine the discrepancy regularization loss ratio. λ_3 is the L2-regularization coefficient for reducing overfitting.

5 Experimental Results

Below, we aim to answer three study questions: (RQ1) How does DSCKG perform compared to other recommendation models? (RQ2) How do hyperparameters affect DSCKG performance? (RQ3) What impact do the various components have on DSCKG?

Table 1. Statistics for the three KGR datasets.

Datasets	Last.FM	Dianping-Food	MovieLens-1M
Users	1872	2298698	6036
Items	3846	1362	2445
Interactions	42346	23416418	753772
Entites	9366	28115	182011
Relations	60	7	12
Triples	15518	160519	1241995

5.1 Dataset Preparation

We evaluate DSCKG on three real-world KGR datasets. (1) **Last.FM**[1] is a online music listening dataset compiled from about 2,000 Last.FM online music systems. (2) **Dianping-Food**[2] is a business dataset from Meituan, consisting

[1] grouplens.org/datasets/hetrec-2011/.
[2] https://www.dianping.com/.

of over 10 million different interactions between about 2 million users and 1,000 restaurants. (3) **MovieLens-1M**[3] is a classic movie recommendation dataset that contains about 1 million explicit ratings. The KGs build construction tool uses Microsoft and Meituan, and all data pre-processing is consistent with the baseline methods [9,10]. (See Table 1 for statistics on the three datasets)

5.2 Competitors

To illustrate the validity of our model, we have chosen eight baselines, covering:

- **CKE** [21]: It's an embedding-based KGR approach that incorporates multi-modal knowledge in a single framework.
- **PER** [20]: It's a classical path-based approach that captures long-distance connections between users and items via meta-paths.
- **RippleNet** [9]: It's an embedding- and propagation-based approach that fully simulates the users' preferences in KGs. And treats users' preferences as Ripple that captures high-order attributes on the KGs.
- **KGCN** [11]: It's a GNNs-based method which enriches item embed-dings via iteratively integrating nearby information.
- **KGNN-LS** [10]: It's a GNNs-based method which enriches item em-bedding via label smoothing regularization.
- **KGAT** [14]: It's a GNNs-based method that iteratively integrates neighbors on the CKG via the attention mechanism to acquire the user/item representation.
- **CKAN** [16]: It's a GNNs-based method which independently propagate CF signals and KGs signals.
- **KGIN** [15]: It's a SOTA GNNs-based approach which combines intentions and KG relations with a finer long-range semantics of intentions and relational paths for node representation.

5.3 Experimental Settings and Evaluation Metrics

We strictly follow the data split ratio of 6:2:2 to compare with all models. We implement our DSCKG in Pytorch and carefully tune the core parameters. We use the Adam optimizer and the Xavier algorithm to optimize and initialize the model parameters. The embedding dimension size in the range of {8, 16,..., 256}. During KGs and collaborative view, the depth of GNNs layer and Light-GCN layer are both adjusted among {1, 2, 3, 4}. For temperature coefficient τ, we search for the number within {0.5, 0.4,..., 0.1}, the self-supervised learning weight is tuned amongst {1e−8, 1e−7,..., 1e−4}. The optimal settings of hyper-parameters in all methods were studied by empirical studies or original papers. We use AUC, F1, and Recall@50,100 as evaluation metrics.

[3] https://grouplens.org/datasets/movielens/1m/.

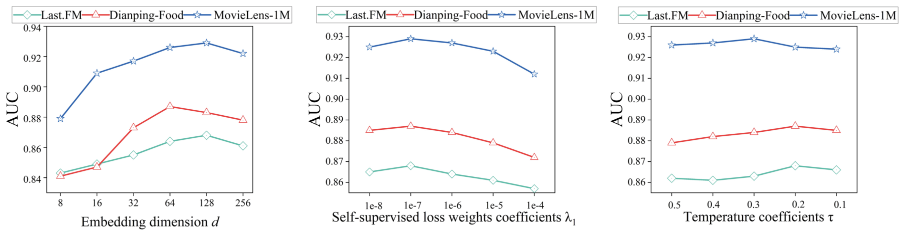
Table 2. Results on two KGR task. *Indicates the significant improvement between our DSCKG and the best performed baseline with $p < 0.05$.

Method	Last.FM				Dianping-Food				MovieLens-1M			
	AUC	F1	R@50	R@100	AUC	F1	R@50	R@100	AUC	F1	R@50	R@100
CKE	0.747	0.674	0.181	0.297	0.812	0.741	0.305	0.439	0.906	0.802	0.375	0.522
PER	0.641	0.603	0.117	0.177	0.766	0.697	0.256	0.355	0.712	0.667	0.171	0.251
RippleNet	0.776	0.702	0.235	0.356	0.863	0.783	0.392	0.545	0.919	0.842	0.317	0.460
KGCN	0.802	0.708	0.298	0.345	0.845	0.774	0.331	0.441	0.909	0.836	0.358	0.512
KGNN-LS	0.805	0.722	0.271	0.349	0.852	0.778	0.342	0.488	0.914	0.841	0.359	0.513
KGAT	0.829	0.742	0.365	0.451	0.846	0.785	0.389	0.561	0.914	0.844	0.385	0.533
CKAN	0.842	0.769	0.342	0.432	0.878	0.802	0.413	0.574	0.915	0.842	0.395	0.540
KGIN	0.848	0.760	0.367	0.462	0.879	0.799	0.415	0.577	0.919	0.844	0.402	0.544
DSCKG	**0.868***	**0.776***	**0.375***	**0.472***	**0.887***	**0.807***	**0.421***	**0.584***	**0.929***	**0.858***	**0.417***	**0.552***
Gain%	**2.35%**	**0.91%**	**2.17%**	**2.16%**	**0.91%**	**0.87%**	**1.44%**	**1.21%**	**1.08%**	**1.65%**	**3.73%**	**1.47%**

5.4 Overall Performance Analysis (RQ1)

To answer the RQ 1, we report the experimental results of DSCKG with eight baselines in Table 2. From the analysis of the experimental results, we have the following observations.

– **Embedding- and path-based methods have unsatisfactory performance.** We first analyze three conventional KGR approaches. PER is typical of meta-path-based methods. CKE utilizes only 1-order knowledge entities as complementary information for recommendations. RippleNet combines path and embedding approaches. We can find that they all achieve unsatisfactory results. We can conclude that 1) path-based methods require human-defined paths, which are often difficult to optimize. 2) Embedding-based methods cannot capture long-range connectivity information. 3) The above methods cannot adaptively capture high-order entity information.
– **GNNs-based KGR methods are dominant.** Analyzing several GNNs-based KGR models, their performance confirms that capturing high-order knowledge and using attention mechanisms effectively enhance recommendation. This inspires us that adaptive mining of long-range information on graphs is essential to boost recommendation effectiveness.
– **Our proposed DSCKG achieves the best performance.** DSCKG performs best in two recommendation tasks, which demonstrates that DSCKG is highly competitive. The main performance improvements are due to: 1) The foundational role of GNNs and attention mechanisms for extracting high-order knowledge; 2) The powerful of feature interaction modelling; 3) The usefulness of self-supervised learning for augmenting node representation learning ability.



(a) Embedding dimension. (b) Self-supervised loss weights. (c) Temperature coefficents.

Fig. 2. Parameter analysis.

5.5 Sensitivity Analysis (RQ2)

Next, we investigate the impact of key hyperparameters on the performance of our DSCKG, including embedding dimension (d), self-supervised loss weights coefficients (λ_1), temperature coefficients (τ), and two view depths (L/L').

Figure 2(a) shows the effect of embedding dimension d. Our DSCKG achieves the best performance with the settings of d = 128, 64, 128, indicating that it encodes the user/item and KGs entity information well. We also find that the performance of our model first increases as d grows and then decreases. We analyze the reason may be that too small dimensions do not have enough ability to capture the necessary information, while too large sizes introduce unnecessary noise and reduce the generalization ability of the model.

Figure 2(b) displays the effect of self-supervised loss weights coefficients λ_1. We have the following observation: The model performance reaches the best when λ_1 is taken as uniformly 1e−7. To analyze the cause, when self-supervised loss weight is set too large, it misleads the direction of the recommendation model optimization and leads to more prominent side effects. Conversely, when the value of λ_1 is taken too small, self-supervised learning fails to achieve significant optimization results and even hinders the correct representation of the model. Therefore, choosing the suitable λ_1 value can augment the node representation learning ability and thus improve the recommendation performance.

Figure 2(c) summarizes the effect of temperature coefficients τ. We consistently conclude that the model performance is best when τ is taken as [0.2, 0.3]. After analysis, increasing the value of τ (e.g., 0.5) will lead to poorer model performance, indicating that self-supervised learning cannot correctly distinguish between positive/negative samples. Conversely, when τ is fixed at a too small value (e.g., 0.1), the model performance is compromised, suggesting that a smaller value of τ could misdirect the gradient optimization direction. This is consistent with the findings of previous work [18].

To investigate the effect of dual-view depth on model performance, we vary L and L' in the range {1, 2, 3, 4} and show the performance comparison of the three datasets in Table 3. We have the following findings: The model performance is optimal when $L = 2, 1, 2$ and $L' = 2, 1, 2$. This indicates that 1 or 2 orders are sufficient to aggregate enough neighbors or remote knowledge for the purpose of enriching the information in different views, whether it is collaborative view or KGs structural view. Conversely, if the model depth is too large, the well-known oversmoothing and the introduction of irrelevant noise will occur, leading to degradation of the model performance.

Table 3. AUC results of DSCKG with depth in dual-view.

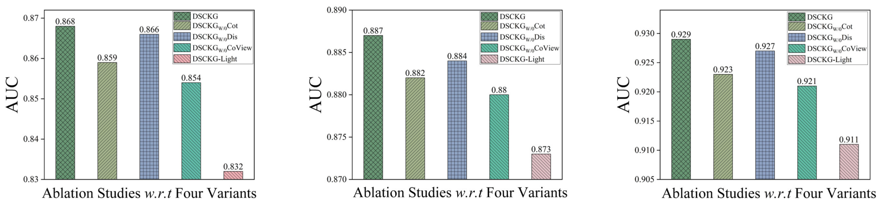
Depth in KGs structural view	1	2	3	4
Last.FM	0.864	**0.868**	0.865	0.863
Dianping-Food	**0.887**	0.883	0.881	0.878
MovieLens-1M	0.926	**0.929**	0.925	0.922
Depth in collaborative view	1	2	3	4
Last.FM	0.863	**0.868**	0.866	0.867
Dianping-Food	**0.887**	0.886	0.883	0.882
MovieLens-1M	0.927	**0.929**	0.924	0.923

5.6 Ablation Studies (RQ3)

Lastly, as shown in Fig. 3, here we examine the contribution of the main components of our model to the final performance by comparing DSCKG with the following four variants.

Variants. 1) DSCKG$_{W/O}$Cot: Remove self-supervised co-training mode. 2) DSCKG$_{W/O}$Dis: Remove discrepancy regularization term. 3) DSCKG$_{W/O}$ CoView: Remove the collaborative view encoder completely. 4) DSCKG-Light: Retain only the backbone networks.

Conclusion. We have the following observations: 1) Removing the self-supervised learning module significantly reduces model performance, indicating that self-supervised co-training mode is critical for node representation learning. 2) Removing the discrepancy regularization term slightly reduces model performance. This reflects the advantage and contribution of the discrepancy regularization to encourage diversity in the representation of collaborative view nodes. 3) Removing the collaborative view modeling means that neither the self-supervised learning nor the discrepancy constraints work properly. We find noticeable performance drop, which justifies the construction of dual views to extract interaction information from different sources and the basis for self-supervised learning. 4) To study in-depth the contribution of our proposed feature interaction (DuCNNs) module, we create a variant called DSCKG-light based on DSCKG$_{W/O}$CoView, i.e., only the backbone network is kept. When DuCNNs are further erased, a significant performance degradation occurs, indicating that mining finer-grained feature interaction signals is indispensable for the KGR model and plays a decisive role in performance improvement.



a) Last.FM b) Dianping-Food c) MovieLens-1M

Fig. 3. Ablation studies.

6 Conclusion and Future Work

In this paper, we aim to explore the knowledge graph recommendation task. Specifically, to mitigate the neglect of feature interaction modeling and the lack of supervised signal problems of conventional KGR methods, we propose a novel DSCKG framework to deal with the above issues via dual-core convolutional neural networks and a self-supervised co-training mode. Extensive experiments validate the effectiveness of our DSCKG framework. There are two directions for future efforts: 1) Design a more reasonable self-supervised paradigm [1,4] and 2) Transfer our novel self-supervised co-training paradigm to other domains (e.g., knowledge tracking [12,13] and session recommendation [23,24]).

Acknowledgement. This work is supported by the Industrial Support Project of Gansu Colleges (2022CYZC-11), the Gansu Natural Science Foundation Project (21JR7RA114), the National Natural Science Foundation of China (62276073, 61762078) and NWNU Teachers Research Capacity Promotion Plan (NWNU-LKQN2019-2). NWNU Graduate Research Project Funding Program (2021KYZZ02103).

References

1. Gao, Z., Ma, H., Zhang, X., Wu, Z., Li, Z.: Co-contrastive self-supervised learning for drug-disease association prediction. In: Proceedings of PRICAI (2022)
2. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation. In: Proceedings of SIGIR (2020)
3. Jiang, Y., Ma, H., Zhang, X., Li, Z., Chang, L.: An effective two-way metapath encoder over heterogeneous information network for recommendation. In: Proceedings of ICMR (2022)
4. Li, Q., Ma, H., Zhang, R., Jin, W., Li, Z.: Co-contrastive learning for multi-behavior recommendation. In: Proceedings of PRICAI (2022)
5. Ma, Y., He, Y., Zhang, A., Wang, X., Chua, T.-S.: CrossCBR: cross-view contrastive learning for bundle recommendation. In: Proceedings of SIGKDD (2022)
6. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv:1807.03748 (2018)

7. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of WSDM (2018)
8. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
9. Wang, H., et al.: RippleNet: propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of CIKM (2018)
10. Wang, H., et al.: Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In: Proceedings of SIGKDD (2019)
11. Wang, H., Zhao, M., Xie, X., Li, W., Guo, M.: Knowledge graph convolutional networks for recommender systems. In: Proceedings of WWW (2019)
12. Wang, W., Ma, H., Zhao, Y., Li, Z., He, X.: Tracking knowledge proficiency of students with calibrated q-matrix. Exp. Syst. Appl. (2022)
13. Wang, W., Ma, H., Zhao, Y., Yang, F., Chang, L.: SEEP: semantic-enhanced question embeddings pre-training for improving knowledge tracing. Inf. Sci. (2022)
14. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S.: KGAT: Knowledge graph attention network for recommendation. In: Proceedings of WWW (2019)
15. Wang, X., et al.: Learning intents behind interactions with knowledge graph for recommendation. In: Proceedings of the Web Conference (2021)
16. Wang, Z., Lin, G., Tan, H., Chen, Q., Liu, X.: CKAN: collaborative knowledge-aware attentive network for recommender systems. In: Proceedings of SIGIR (2020)
17. Wei, Y., Ma, H., Zhang, R., Li, Z., Chang, L.: Exploring implicit relationships in social network for recommendation systems. In: Proceedings of PAKDD (2021)
18. Wu, J., et al.: Self-supervised graph learning for recommendation. In: Proceedings of SIGIR (2021)
19. Xia, X., Yin, H., Yu, J., Shao, Y., Cui, L.: Self-supervised graph co-training for session-based recommendation. In: Proceedings of CIKM (2021)
20. Yu, X., et al.: Personalized entity recommendation: a heterogeneous information network approach. In: Proceedings of WSDM (2014)
21. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of SIGKDD (2016)
22. Zhang, R., Ma, H., Li, Q., Wang, Y., Li, Z.:. Fire: knowledge-enhanced recommendation with feature interaction and intent-aware attention networks. Appl. Intell. (2022)
23. Zhang, X., Ma, H., Gao, Z., Li, Z., Chang, L.: Exploiting cross-session information for knowledge-aware session-based recommendation via graph attention networks. Int. J. Intell. Syst. (2022)
24. Zhang, X., Ma, H., Yang, F., Li, Z., Chang, L.: Cross-view contrastive learning for knowledge-aware session-based recommendation. In: Proceedings of PRICAI (2022)