Knowledge Graph Contrastive Learning for Recommendation

Yuhao Yang
University of Hong Kong
yuhao-yang@outlook.com

Chao Huang*
University of Hong Kong
chaohuang75@gmail.com

Lianghao Xia
University of Hong Kong
aka_xia@foxmail.com

Chenliang Li
Wuhan University
cllee@whu.edu.cn

arXiv:2205.00976v2 [cs.IR] 21 May 2022

ABSTRACT

Knowledge Graphs (KGs) have been utilized as useful side information to improve recommendation quality. In those recommender systems, knowledge graph information often contains fruitful facts and inherent semantic relatedness among items. However, the success of such methods relies on the high quality knowledge graphs, and may not learn quality representations with two challenges: i) The long-tail distribution of entities results in sparse supervision signals for KG-enhanced item representation; ii) Real-world knowledge graphs are often noisy and contain topic-irrelevant connections between items and entities. Such KG sparsity and noise make the item-entity dependent relations deviate from reflecting their true characteristics, which significantly amplifies the noise effect and hinders the accurate representation of user's preference.

To fill this research gap, we design a general Knowledge Graph Contrastive Learning framework (KGCL) that alleviates the information noise for knowledge graph-enhanced recommender systems. Specifically, we propose a knowledge graph augmentation schema to suppress KG noise in information aggregation, and derive more robust knowledge-aware representations for items. In addition, we exploit additional supervision signals from the KG augmentation process to guide a cross-view contrastive learning paradigm, giving a greater role to unbiased user-item interactions in gradient descent and further suppressing the noise. Extensive experiments on three public datasets demonstrate the consistent superiority of our KGCL over state-of-the-art techniques. KGCL also achieves strong performance in recommendation scenarios with sparse user-item interactions, long-tail and noisy KG entities. Our implementation codes are available at https://github.com/yuh-yang/KGCL-SIGIR22.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommendation; Knowledge Graph; Self-Supervised Learning

*Chao Huang is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR'22, July 11–15, 2022, Madrid, Spain
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3532009

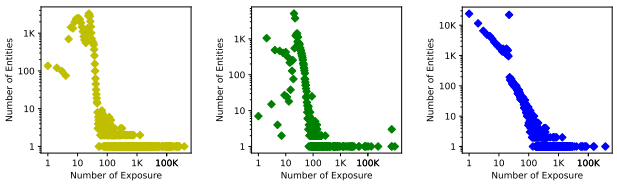
ACM Reference Format:
Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge Graph Contrastive Learning for Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3532009

1 INTRODUCTION

Recommender systems have increasingly become an integral installation for suggesting interested items to users and alleviating information overloading in many online services, ranging from E-commerce platforms [38], video-sharing sites [20] to online advertising [9]. Among various techniques, Collaborative Filtering (CF) frameworks become effective solutions to predict users' preferences, based on the rationale that users with similar interaction behaviors may share similar interests for items [12, 18, 26].

In recent years, the prevalent collaborative filtering paradigms have evolved from matrix factorization (MF) to neural network-based techniques for latent user and item embedding projection, such as Autoencoder-based approaches (*e.g.,* Autorec [28]), attentive CF mechanisms (*e.g.,* ACF [4]), as well as recently developed CF models built upon graph convolutional architectures (*e.g.,* LightGCN [11]). However, even with the modeling of complex user-item interaction patterns, most CF-based recommendation methods still suffer from the data scarcity issue for users who have not yet interacted with sufficient items [7, 14, 31]. To overcome such data sparsity problem, Knowledge Graphs (KGs) serving as useful external sources have been incorporated into the recommender system to enhance the user and item representation process, by encoding additional item-wise semantic relatedness [15, 39, 47].

Existing KG-enhanced methods can be roughly categorized into three groups. Particularly, some studies [3, 51] bridge the knowledge graph learning with user-item interaction modeling, through adopting the transition-based entity embedding schemes (*e.g.*,, TransE [2], TransR [19]) to generate prior item embeddings. To improve KG-enhanced recommender systems in capturing high-order KG connectivity, some path-based models [35, 41, 52] aim to construct path-guided user-item connections with the incorporated KG entities. Nevertheless, most of those path-based methods involve the design of meta-paths for generating entity-dependent relationships, which requires specific domain knowledge and labor-intensive human efforts for accurate path construction. Motivated by the strength of graph neural networks, one promising recent research line lies in recursively performing information propagation among multi-hop nodes and injecting long-range relational structures, such as KGAT [39], MVIN [29], KHGT [45] and KGIN [40].



(a) Yelp2018 (b) Amazon-Book (c) MIND

Figure 1: Long-tail entity distributions in real-world KGs.

Despite their effectiveness in some scenarios, we argue that the effectiveness of existing KG-aware recommendation methods largely relies on the high quality input knowledge graphs and are vulnerable to noise perturbation. However, in practical scenarios, knowledge graphs are often sparse and noisy by exhibiting long-tail entity distribution and containing topic-irrelevant connections between items and entities [24, 34].

We report distributions of KG entities collected from three real-world datasets in Figure 1 to illustrate the long-tail issue in KGs. In this figure, the y-axis represents the number of entities corresponding to the exposure count in the x-axis. Obviously, across datasets from different platforms, *i.e.,* food, book and news, the majority of KG entities exhibit long-tail phenomenon. Since it requires enough triplets (h, r, t) linked to an entity to accurately model semantic transitions in the KG by employing Trans algorithms [2, 19], which brings challenge for accurately capturing item-wise relatedness. Additionally, topic-irrelevant entity connections are ubiquitous in KGs. We present a motivating example for news recommendation shown in Figure 2, the key entity *Zack Wheeler* extracted from the news item is a famous baseball pitcher for the Philadelphia Phillies in the organization of Major League Baseball (MLB). However, we can notice that *Zack Wheeler* is linked with two item semantic-irrelevant "noisy" entities, *i.e., Smyrna, GA* and *UCL Reconstruction*. While *Zack Wheeler* was born in *Smyrna, GA* and he underwent a surgery for Ulnar Collateral Ligament (UCL) reconstruction before, these two entities are less relevant to the main topic of this news with the focus on the recent sports news.

We unify the aforementioned problems as the KG noise issue. Such data noise issue will impair the quality of item representation from two perspectives: i) From the local view, directly aggregating information from those low-quality entities will bring noise in preserving key semantics of items from their neighboring entities. ii) From the global view, the information aggregation over the knowledge graph is easily towards over-smoothed, since the overwhelming information can be propagated to the target node through some popular entities (*e.g.,* location names). For example, other persons who are also born in *Smyrna, GA* can be connected to the professional baseball pitcher–*Zack Wheeler*. Hence, it is a necessity to endow the knowledge graph-enhanced CF paradigm with the capability of effectively connection denoising, so as to distill the true underlying preference of target users with representations invariant to noise disturbance.

Contribution. In light of the aforementioned limitations and challenges, we propose a general Knowledge Graph Contrastive Learning framework (KGCL) for recommendation. Specifically, to handle the relation heterogeneity in knowledge graph, we first propose a relation-aware knowledge aggregation mechanism to capture the

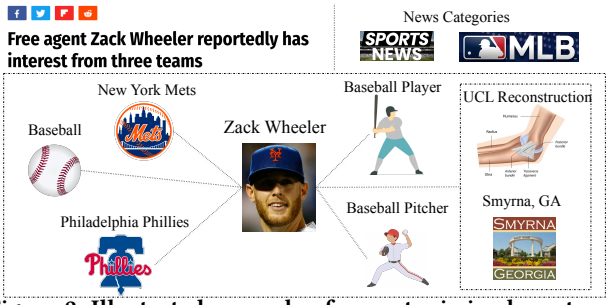


Figure 2: Illustrated example of news topic-irrelevant entities extracted from the knowledge graph of MIND dataset.

entity- and relation-dependent contextual signals for the first-stage item representation. Then, we develop a cross-view contrastive learning schema which bridges the knowledge graph denoising with the user-item interaction modeling, such that the external item semantic relatedness can be leveraged to guide the data augmentation with cross-view self-supervised signals. The designed cross-view contrastive learning schema suppresses KG noise via performing KG contrastive learning, and exploits external signals from the process to measure the bias of item representations affected by KG noise. The signals serve as the guidance to the user-item graph contrastive learning view, keeping useful graph structures and involve less noise.

KGCL takes inspirations from the knowledge graph learning and self-supervised data augmentation, to incorporate the knowledge graph context to guide the model in refining user/item representations with new knowledge-aware contrastive objectives. In our framework, our joint contrastive learning network learns to drop irrelevant KG triplets and items based on the knowledge graph structural consistency for robust user preference learning. Due to the model-agnostic property of our KGCL, it can be plugged into various graph neural recommendation models. In KGCL, the knowledge graph-guided contrastive learning model and graph neural CF architecture are jointly optimized in an end-to-end manner.

In summary, our contributions are highlighted as follows:

- This work introduces the idea of integrating the knowledge graph learning with user-item interaction modeling under a joint self-supervised learning paradigm, to improve the robustness and alleviate the data noise and sparsity issues for recommendation.
- We present a general KGCL, a knowledge graph-guided topological denoising framework, offering cross-view self-discrimination supervision signals with knowledge-aware contrastive objective. We also provide theoretical analysis to justify the benefits brought by the integrative learning objective.
- We conduct diverse experiments on three public datasets and the proposed KGCL consistently outperforms various state-of-the-art recommendation methods across different settings. Further ablation analysis justifies the rationality of our key components.

2 PRELIMINARIES

This section introduces key notations used throughout the paper and formalize our studied task. We consider a typical recommendation scenario with a user set $\mathcal{U}$ and an item set $\mathcal{I}$. Individual user and item is denoted as u ($u \in \mathcal{U}$) and i ($i \in \mathcal{I}$), respectively. We define the user-item interaction matrix $\mathcal{Y} \in |\mathcal{U}| \times |\mathcal{I}|$ to represent

the consumption behaviors of users over different items. In matrix $\mathcal{Y}$, the element $y_{u,i} = 1$ given that user u has adopted item i before (*e.g.*, click, review or purchase) and $y_{u,i} = 0$, otherwise.

User-Item Interaction Graph. Based on the matrix $\mathcal{Y}$, we first construct the user-item interaction graph $\mathcal{G}_u = \{\mathcal{V}, \mathcal{E}\}$, where the node set $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ and edge (u, i) is generated in $\mathcal{G}$ if $y_{u,i} = 1$.

Knowledge Graph. We let $\mathcal{G}_k = \{(h, r, t)\}$ represent the knowledge graph which organizes external item attributes with different types of entities and corresponding relationships. Specifically, each entity-relation-entity triplet (h, r, t) characterizes the semantic relatedness between the head and tail entity h and t with the relation r, such as the triplet for movie recommendation (Titanic, Directed by, James Cameron), and venue recommendation (McDonald's, Located in, Chicago). Such information incorporates fruitful facts and connections among items as side information to improve the modeling of user preference for recommendation.

Having constructed user interaction behaviors and item knowledge, we seek to leverage the item knowledge information to assist the user interest learning. However, real-world knowledge graphs are often noisy and involve item-irrelevant entities as we described before. In such cases, not all entities and relations are useful for learning appropriate item characteristics. In most existing knowledge-aware recommender systems, messages aggregated from "noisy"entiries and relations may heavily impact the quality of item representation, which notably limits the effectiveness of KG-enhanced user preference modeling. To tackle this challenge, this work exploits the potential of knowledge graph topology denoising to distill informative guidance for user and item representations.

Task Formulation. We formally describe our task as follows: **Input**: user-item interaction data $\mathcal{G}_u = \{\mathcal{V}, \mathcal{E}\}$ and item knowledge graph data $\mathcal{G}_k = \{(h, r, t)\}$. **Output**: the learned function $\mathcal{F} = (u, v|\mathcal{G}_u, \mathcal{G}_k, \Theta)$ that forecasts the items user u ($u \in \mathcal{U}$) would like to interact with, where Θ denotes the model parameters.

3 METHODOLOGY

We present the overall architecture of KGCL in Figure 3. Technical details are discussed in following sub-sections.

3.1 Relation-aware Knowledge Aggregation

3.1.1 **Knowledge Aggregation with Relation Heterogeneity**. Inspired by the graph attention mechanisms in [32, 39, 45], we first design a relation-aware knowledge embedding layer to be reflective of relation heterogeneity over knowledge graph connection structures. To address the limitation of manually design of path generation on knowledge graphs, our KGCL projects entity- and relation-dependent context into specific representations with parameterized attention matrix. Towards that, we built our message aggregation mechanism between the item and its connected entities in $\mathcal{G}_k$, for generating knowledge-aware item embeddings based on the heterogeneous attentive aggregator shown as follows:

$$\mathbf{x}_i = \mathbf{x}_i + \sum_{e \in \mathcal{N}_i} \alpha\left(e, r_{e,i}, i\right) \mathbf{x}_e$$

$$\alpha\left(e, r_{e,i}, i\right) = \frac{\exp\left(LeakyReLU\left(\mathrm{r}_{e,i}^{\top} \mathrm{W}\left[\mathbf{x}_e \| \mathbf{x}_i\right]\right)\right)}{\sum_{e \in N_i} \exp\left(LeakyReLU\left(\mathrm{r}_{e,i}^{\top} \mathrm{W}\left[\mathbf{x}_e \| \mathbf{x}_i\right]\right)\right)} \quad (1)$$

where N_i is the neighboring entities of item i based on different types of relations $r(e, i)$ in knowledge graph $\mathbf{G}_k$. Here, the embedding of item and entity is denoted as $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{x}_e \in \mathbb{R}^d$, respectively. $\alpha\left(e, r_{e,i}, i\right)$ represents the estimated entity- and relation-specific attentive relevance during the knowledge aggregation process. In particular, $\alpha\left(e, r_{e,i}, i\right)$ encodes the distinct semantics of relationships between item i and entity e. $\mathrm{W} \in \mathbb{R}^{d \times 2d}$ represents the parametric weight matrix customized to the input item and entity representations. *LeakyReLU* activation function is adopted for non-linear transformation.

3.1.2 **Semantic Representation Enhancement**. Additionally, to further enhance the multi-relational semantic representation space for entity-item dependencies, we perform the alternative training between our relation-aware knowledge aggregator and TransE [2]. The general idea of this translation-based knowledge graph embedding is to make the summation of head and relation embedding $\mathbf{x}_h$ and $\mathbf{x}_r$ as close as the tail representation $\mathbf{x}_t$. Here, we define $f_d(\cdot)$ to represent the L_1 norm-based similarity measurement function between embedding vectors, *i.e.*, $f_d = \|\mathbf{x}_h + \mathbf{x}_r - \mathbf{x}_t\|$. Formally, the translation-based optimized loss $\mathcal{L}_{TE}$ is shown below:

$$\mathcal{L}_{TE} = \sum_{(h,r,t,t') \in \mathcal{G}_k} -\ln \sigma\left(f_d(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_{t'}) - f_d(\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t)\right) \quad (2)$$

The negative sample t' is generated by randomly replacing the tail t for the observed triplets (h, r, t) from the knowledge graph $\mathcal{G}_k$.

3.2 Knowledge Graph Augmentation

3.2.1 **Augmentation on Knowledge Graph Structures**. Motivated by the recent success of data augmentation techniques with contrastive learning in CV/NLP tasks, *e.g.*, image analysis [33] and machine translation [27], we propose to bridge the knowledge graph embedding and the contrastive learning paradigm with auxiliary self-supervised signals. At the core of contrastive learning is to maximize the mutual information between augmented views and regularize the embedding learning with contrastive objectives.

In our KGCL framework, we propose to generate different views of knowledge graph structures for contrastive learning through the entity-wise self-discrimination. In particular, we adopt stochastic data augmentation scheme over the input knowledge graph to generate two correlated data views. Then, the knowledge graph structural consistency of individual item is derived to be reflective of item-wise invariance to knowledge noise perturbation. Towards this end, we hence devise the data augmentation operator $\eta(\cdot)$ on the knowledge graph structure with two stochastic selections $\eta_1(\mathcal{G}_k)$ and $\eta_2(\mathcal{G}_k)$, which can be formally presented as follows:

$$\eta_1(\mathcal{G}_k) = ((e, r, i) \odot \mathbf{M}_k^1),\ \eta_2(\mathcal{G}_k) = ((e, r, i) \odot \mathbf{M}_k^2) \quad (3)$$

where $(e, r, i) \in \mathcal{G}_k$ represents the knowledge triplet between items and their dependent entities. Here, we define masking vectors $\mathbf{M}_k^1$, $\mathbf{M}_k^2 \in \{0, 1\}$ as the binary indicators with the probability p_k, to denote whether the specific knowledge triplet is selected or not during the sampling. By doing so, we can generate knowledge subgraph with different augmented structural views. The objective of our knowledge graph augmentation scheme is to identify items which are less sensitive to structure variation, and more tolerant to the connections with noisy entities. Such identified items are less

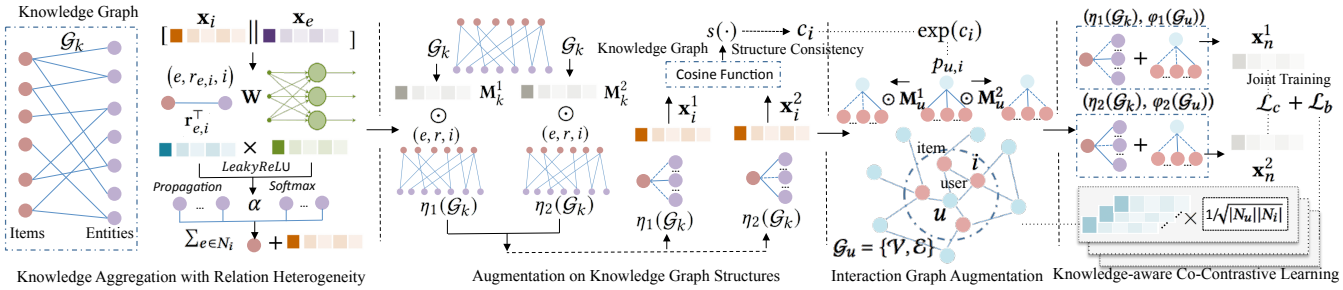


Figure 3: The overall architecture of our proposed KGCL. Knowledge-aware co-contrastive learning with augmentation functions on both knowledge graph $\eta(\cdot)$ and user-item interaction graph $\varphi(\cdot)$. Our contrastive objective $\mathcal{L}_c$ is jointly optimized with main embedding space shared by the knowledge graph aggregation and graph-based CF encoder.

ambiguous in terms of their characteristics and are more helpful to capture the preference of the correlated users.

3.2.2 **Agreement between Augmented Structural Views**. After performing the augmentation on knowledge graph structures, we obtain two knowledge graph dependency views with operators $\eta_1(\cdot)$ and $\eta_2(\cdot)$. Inspired by the investigation of graph consistency in [16, 55], to explore the agreement property of each item based on the augmented views, we define knowledge graph structure consistency c_i of item i with the agreement between the representations encoded from different views as follows:

$$c_i = s\left(f_k\left(\mathrm{x}_i, \eta_1(\mathcal{G}_k)\right), f_k\left(\mathrm{x}_i, \eta_2(\mathcal{G}_k)\right)\right) \tag{4}$$

Here, f_k represents the relation-aware knowledge aggregation scheme (defined in Eq 1), to generate item embeddings $\mathbf{x}_i^1$ and $\mathbf{x}_i^2$ corresponding to different augmented structure views $\eta_1(\mathcal{G}_k)$ and $\eta_2(\mathcal{G}_k)$. $s(\cdot)$ denotes the cosine function to estimate the similarity between $\mathbf{x}_i^1$ and $\mathbf{x}_i^2$. Based on the above definitions, we can notice that if an item achieves a higher structure consistency score c_i, it is less sensitive to the topological information changes. Therefore, if item i is affected more by KG noise than item i', it is more likely that $c_i < c_{i'}$. Such derived knowledge structure consistency property of each item can be adopted as the guidance, so as to against both the knowledge graph dependency and the user-item interaction noise with auxiliary self-supervised signals.

3.3 Knowledge-Guided Contrastive Learning

We integrate our knowledge graph augmentation schema with the graph contrastive learning paradigm, with the aim of improving the representation ability of graph-based collaborative filtering in terms of model accuracy and robustness. To effectively transfer useful item external knowledge in learning of user preference, we design two contrastive representation spaces for user-item interactions. In such contrastive learning framework, the denoising item knowledge can be leveraged to guide the user and item representation and alleviate the sparsity of supervision signals.

3.3.1 **Interaction Graph Augmentation Mechanism**. While the recent proposed self-supervised recommendation model SGL [44] performs data augmentation on user-item interaction graph, the purely randomly dropout operations limits its effectiveness in keeping useful interactions for contrastive learning.

To mitigate this limitation, we leverage the estimated knowledge graph structure consistency of items to guide the data augmentation over the user-item interaction graph $\mathcal{G}_u = \{\mathcal{V}, \mathcal{E}\}$. The rationale behind our knowledge-guided graph contrastive learning is to identify interactions which are more useful to characterize user preference with less bias information. To be specific, the items with higher KG structure consistency scores will involve less noise and contribute more to the modeling of user's real interests. In accordance with our knowledge-guided augmentation, we incorporate the derived item-specific KG structure consistency c_i into our operator on the user-item interaction graph with the following formulas:

$$\begin{aligned} w_{u,i} &= \exp(c_i);\ p'_{u,i} = \max\left(\frac{w_{u,i} - w^{min}}{w^{max} - w^{min}}, p_\tau\right) \\ p_{u,i} &= p_a \cdot \mu_{p'} \cdot p'_{u,i} \end{aligned} \tag{5}$$

where $p_{u,i}$ represents the estimated probability to dropout the interaction edge between user u and item i. $w_{u,i}$ represents the influence degree of item i over user u, which is proportional to the corresponding structure consistency score of c_i. We further perform the min-max normalization on $w_{u,i}$ with the truncation probability p_τ, to alleviate the low value effect. After that, the intermediate variable $p'_{u,i}$ is obtained and integrated with the mean value $\mu_{p'}$ to derive the value of dropout probability $p_{u,i}$. Here, p_a controls the strength of mean-based influence. With the probability $p_{u,i}$, we further generate two masking vectors $\mathbf{M}_u^1, \mathbf{M}_u^2 \in \{0, 1\}$ based on the Bernoulli distribution [23]. After that, $\mathbf{M}_u^1, \mathbf{M}_u^2 \in \{0, 1\}$ are applied to the user-item interaction graph $\mathcal{G}_u$ as follows:

$$\varphi(\mathcal{G}_u) = (\mathcal{V}, \mathbf{M}_u^1 \odot \mathcal{E}),\ \varphi(\mathcal{G}_u) = (\mathcal{V}, \mathbf{M}_u^2 \odot \mathcal{E}) \tag{6}$$

where $\varphi(\cdot)$ denotes our graph augmentation operator which drops out the user-item interaction in the edge set $\mathcal{E}$ of graph $\mathcal{G}_u$ according to the inferred probability $p_{u,i}$.

3.3.2 **Knowledge-aware Co-Contrastive Learning**. Different from most existing contrastive learning models (*e.g.*, SGL [44], GraphCL [49]) which directly performs transformation on structure-level augmentations (*e.g.*, node dropping out or edge edge perturbation), we incorporate item knowledge semantics into a co-contrastive learning architecture. Our KGCL aims to improve the model robustness with augmented self-supervision signals, by disturbing the graph structures from the views of both item semantics and user behavioral patterns. In our co-contrastive learning paradigm, we integrate our designed graph augmentation operators $\eta(\mathcal{G}_k)$ and $\varphi(\mathcal{G}_u)$ to create two contrastive views, which enables view-specific encoders collaboratively supervise with each other.

Specifically, given the obtained augmented knowledge subgraphs through $\eta_1(\mathcal{G}_k)$ and $\eta_2(\mathcal{G}_k)$, we further separately corrupt the user-item interaction graph guided by the derived knowledge structure consistency of items ($c_i, i \in \mathcal{I}$) based on $\varphi_1(\mathcal{G}_u)$ and $\varphi_2(\mathcal{G}_u)$. After that, we can create two knowledge-guided corrupted graphs among users, items and entities. Then, we encode the representations of users and items by utilizing the graph-based collaborative filtering framework and the relation-aware knowledge aggregation mechanism. Due to the effectiveness and lightweight architecture of LightGCN [11], we adopt its message propagation strategy to encode the collaborative effects from user-item interactions as below:

$$\mathbf{x}_u^{(l+1)} = \sum_{i \in N_u} \frac{\mathbf{x}_i^{(l)}}{\sqrt{|N_u||N_i|}}; \ \mathbf{x}_i^{(l+1)} = \sum_{u \in N_i} \frac{\mathbf{x}_u^{(l)}}{\sqrt{|N_i||N_u|}} \tag{7}$$

where $\mathbf{x}_u^{(l)}$ and $\mathbf{x}_i^{(l)}$ represents the encoded representations of user u and item i under the l-th graph propagation layer. N_u and N_i denotes the set of user u's interacted items and item i's connected users, respectively. In the graph-structured CF architecture, the high-order collaborative signals can be captured via stacking multiple graph propagation layers. In this encoding pipeline, our designed heterogeneous attentive aggregator (defined in Eq 1) is employed to generate input item feature vector with the preservation of knowledge graph semantics. Such item embeddings are fed into the graph-based CF for representation refinement.

After that, KGCL employs the generated two knowledge-aware graph views, *i.e.,* $(\eta_1(\mathcal{G}_k), \varphi_1(\mathcal{G}_u))$ and $(\eta_2(\mathcal{G}_k), \varphi_2(\mathcal{G}_u))$, to collaboratively supervise each other. Particularly, KGCL performs contrastive learning on view-specific user/item representations ($\mathbf{x}_u^1$, $\mathbf{x}_i^1$) and ($\mathbf{x}_u^2$, $\mathbf{x}_i^2$). For each user or item node, the positive pairs are generated from the two view-specific embeddings of user u or item i based on the self-discrimination ability of node. Negative pairs are the representations of different nodes in both graph views. The contrastive objective $\mathcal{L}_c$ in our KGCL is defined based on the InfoNCE [5] loss as follows:

$$\mathcal{L}_c = \sum_{n \in \mathcal{V}} -\log \frac{\exp(s(\mathbf{x}_n^1, \mathbf{x}_n^2)/\tau)}{\sum_{n' \in \mathcal{V}, n' \neq n} \exp(s(\mathbf{x}_n^1, \mathbf{x}_{n'}^2)/\tau)} \tag{8}$$

where τ is the temperature parameter. We adopt the cosine function $s(\cdot)$ to estimate the similarity of positive pairs and negative pairs. By minimizing the contrastive objective loss $\mathcal{L}_c$, we can achieve the agreement between positive pairs as compared to negative ones.

Joint Training. In the learning process of KGCL, we design a joint embedding space which is shared by the main recommendation task and the auxiliary self-supervised signals. In particular, we further couple the original Bayesian personalized ranking (BPR) recommendation loss with the aforementioned contrastive loss. Firstly, we formally present the employed BPR loss $\mathcal{L}_b$ as follows:

$$\mathcal{L}_b = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{N}_u} \sum_{i' \notin \mathcal{N}_u} -\log \sigma(\hat{y}_{u,i} - \hat{y}_{u,i'}) \tag{9}$$

where $\mathcal{N}_u$ represents the observed interactions of user u. We sample the negative instance from the non-interacted items i' ($i' \notin \mathcal{N}_u$) of user u. $\hat{y}$ is the estimated interaction probability between user u and item i, which is derived with the dot-product as: $\mathbf{x}_u^\top \mathbf{x}_i$. Given above definitions, the integrative optimization loss of our KGCL is:

$$\mathcal{L} = \mathcal{L}_b + \lambda_1 \mathcal{L}_c + \lambda_2 \|\Theta\|_2^2, \tag{10}$$

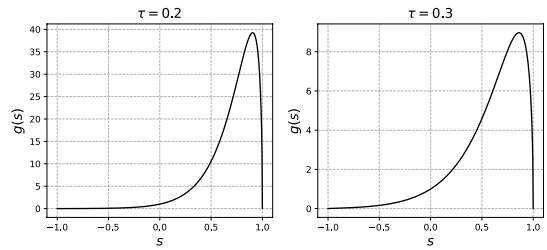


Figure 4: Distribution of gradient function $g(s)$ under $\tau = 0.2$ and $\tau = 0.3$. s is the similarity score between positive and negative instances. Hard negatives significant impact $g(s)$.

where λ_1 and λ_2 denote parameters to determine the strength of self-supervised signals and regularization for the joint loss function. Θ represents the learnable model parameters.

3.4 Model Analysis of KGCL

3.4.1 **Theoretical Discussion of KGCL**. In our KGCL, we leverage the extracted knowledge graph semantics to guide the graph contrastive learning across different structural views, to improve the discrimination ability of hard negatives. Specifically, following the work in [17, 44], given the estimated similarity between one node instance (*i.e.,* user u or item i) and its negative sample v, the obtained contrastive gradient $g(s)$ can be formally presented:

$$g(s) = \sqrt{1 - s^2} \exp\left(\frac{s}{\tau}\right) \tag{11}$$

We show the distribution of $g(s)$ in Figure 4 given that $\tau = 0.2$ and $\tau = 0.3$. As shown in this figure, the hard negatives with high similarity scores (*e.g.,* $0.7 \leq s \leq 0.9$), results in the value of gradient $g(s)$ close to 40. In such case, hard negatives will have a much larger influence on the gradient learning as compared to easy negatives.

In our knowledge-aware contrastive learning paradigm, the derived knowledge graph structural consistency of items is incorporated into the augmentation on user-item interaction graph, to guide the edge dropout operation. By doing so, the discrimination ability of hard negatives in KGCL can be improved from the following aspects: i) items connected with noisy entities are differentiated better with the dropout of noisy triplets *e.g.,* $(e, r_{e,i}, i)$; ii) users interacted with ambiguous items can be modeled with lower similarities. Motivated by the research work in [17, 44], we provide the knowledge-aware contrastive gradient analysis. Particularly, we first define the set of false hard negatives as U_f with the definition:

$$\|\theta_f(U_f, v) - s_0\| < t_0 \tag{12}$$

where the largest value of $g(s)$ is obtained with the maximum point $(s_0, g(s_0))$. $\|\theta_f(U_f, v) - s_0\| = \frac{\sum_{u_f \in U_f} |\theta_f(u_f, v) - x_0|}{|U_f|}$ measures the average similarity between node instances in U_f and the corresponding positive sample v. Here θ_f denotes the similarity estimated with bias involved by the noisy entity-dependent information in knowledge graph. Without the knowledge graph denoising for data augmentations, the similarity between the positive and negative samples is likely to impact the model optimization with large gradient (caused by false hard negatives U_f) according to the distribution curve of $g(s)$:

$$\|\theta_r(U_f, v) - s_0\| \gg t_0 \tag{13}$$

Table 1: Statistics of experimented datasets.

Stats.	Yelp2018	Amazon-Book	MIND
# Users	45,919	70,679	300,000
# Items	45,538	24,915	48,957
# Interactions	1,183,610	846,434	2,545,327
Density Degree	5.7×10^{-4}	4.8×10^{-4}	1.7×10^{-4}
	Knowledge Graph		
# Relations	42	39	90
# Entities	47,472	29,714	106,500
# Triples	869,603	686,516	746,270

Here, $\theta_r(\cdot)$ represents the actual estimated similarity with false hard negatives. In our KGCL, the effect of false hard negatives can be alleviated with our contrastive augmentation functions $\eta(\cdot)$ and $\varphi(\cdot)$. The similarity derived by KGCL is denoted as θ_a. Based on above discussion, we respectively update the similarity derivation for false hard negatives U_f and true hard negatives U_t as follows:

$$\|\theta_r(U_f, v) - s_0\| \geq \|\theta_a(U_f, v) - s_0\| \gg \|\theta_f(U_f, v) - s_0\|$$
$$\|\theta_f(U_t, v) - s_0\| \gg \|\theta_a(U_t, v) - s_0\| \geq \|\theta_r(U_t, v) - s_0\| \quad (14)$$

After enhancing the discrimination ability over hard negatives, we can improve the robustness of knowledge-aware recommender systems with the accurate and helpful gradients for model learning.

3.4.2 **Model Time Complexity Analysis**. We analyze the time complexity from three key components of our KGCL framework. (1) For the knowledge aggregation module, $O(|\mathcal{E}_k| \times d)$ calculations are required to calculate weights α and conduct information aggregation, where $|\mathcal{E}_k|$ denotes the number of relations in the knowledge graph $\mathcal{G}_k$. This module takes additional $O(B_t \times d)$ time for knowledge graph embedding with TransE, where B_t representing the number of training triplets in a batch. (2) Our designed knowledge graph augmentation only takes $O(|\mathcal{E}_k| + |\mathcal{V}| \times d)$ time to derive the KG structure consistency and perturbation. (3) The graph-based collaborative filtering takes $O(|\mathcal{E}| \times d)$ time for user-item interaction modeling. The time complexity to calculate InfoNCE loss $O(B_{id} \times (|\mathcal{U}| + |\mathcal{I}|) \times d)$, where B_{id} is the number of unique users and items within a batch. Based on the above analysis, our KGCL achieves comparable time complexity when competing with state-of-the-art knowledge-aware recommendation models [37, 39].

4 EXPERIMENTS

Extensive experiments are performed to evaluate the performance of our KGCL by answering the following research questions:

- **RQ1**: How does our KGCL perform when competing with different types of recommendation methods?
- **RQ2**: How do different key modules in our KGCL framework contribute to the overall performance?
- **RQ3**: How effective is the proposed KGCL model in alleviating data sparsity and noise issues for recommendation?
- **RQ4**: How is the model interpretation ability of our KGCL?

4.1 Experimental Settings

4.1.1 **Datasets**. We perform experiments on three public datasets collected from different real-life platforms: *Yelp2018* for business venue recommendation, *Amazon-Book* for product recommendation, and *MIND* for news recommendation. Table 1 presents the statistical information of our experimented datasets with different interaction sparsity degrees and knowledge graph characteristics. We follow the similar settings in [39] to construct knowledge graphs for Yelp2018 and Amazon-Book datasets by mapping items into Freebase entities [53]. In our experiments, we only collect entities within two hops since few of the baselines consider modeling multi-hop relations in the KG, and such relations are usually noisy and semantically biased. In our knowledge graphs, various types of entities (*e.g.,* venue category/location, book authors/publisher) are adopted to generate entity-dependent relations. For the news MIND dataset, we follow the data pre-processing strategy in [30] to construct the knowledge graph based on *spacy-entity-linker* tool[1] and Wikidata[2]. Evaluated datasets are available in our released model implementations with the link in the abstract section.

4.1.2 **Evaluation Protocols**. For fair comparison, we employ the all-ranking strategy to be consistent with the settings in [39, 40]. Specifically, for each target user, we regard all his/her non-interacted items as negative samples to infer the preference of this user. For the performance evaluation, two representative metrics: Recall@N and NDCG@N are used to evaluate the accuracy of top-N recommended items [39, 46]. Average evaluation results across all users in the test set are reported with $N = 20$ by default.

4.1.3 **Baselines for Comparison**. We compare KGCL with various lines of recommender systems for performance evaluation.

Conventional Collaborative Filtering Method.

- **BPR** [25]. It is a representative recommendation approach to rank item candidates with a pairwise ranking loss.

MLP-based Neural Collaborative Filtering Framework.

- **NCF** [10]. It utilizes the multiple-layer perceptron to endow the CF architecture with the non-linear feature interaction.

Graph Neural Networks for Collaborative Filtering.

- **GC-MC** [1]. It is built on the graph auto-encoder architecture to capture the interaction patterns between user and item based on the links in the bipartite graph.
- **LightGCN** [11]. This is a state-of-the-art GCN-based recommendation method which simplifies the convolution operations during the message passing among users and items.

Self-Supervised Learning Recommender System.

- **SGL** [44]. This method offers state-of-the-art performance by enhancing the graph-based CF framework with augmented structure-based self-supervised signals.

Embedding-based Knowledge-aware Recommendation.

- **CKE** [51]. This method adopts TransR to encode the items' semantic information and further incorporate it into the denoising auto-encoders for item representation with knowledge base.

Path-based Knowledge-aware Recommendation.

- **RippleNet** [35]. It propagates user preference over the knowledge graph along with the constructed paths rooted at this user. It is a memory-like neural model to improve user representations.

KG-enhanced Recommendation with GNNs.

- **KGCN** [37]. It aims to encode high-order dependent context with respect to the semantic information in KG. At the core of KGCN is to incorporate neighborhood information bias into aggregating message for entity representation.

[1]https://github.com/egerber/spaCy-entity-linker
[2]https://query.wikidata.org/

Table 2: Performance comparison of all methods on Yelp, Amazon and MIND. The superscript $*$ indicates the improvement is statistically significant where p-value < 0.01 level.

Model	Yelp2018		Amazon-book		MIND	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR	5.55%*	0.0375*	12.44%*	0.0658*	9.38%*	0.0469*
NCF	5.35%*	0.0346*	10.33%*	0.0532*	8.93%*	0.0436*
GC-MC	6.88%*	0.0453*	10.64%*	0.0534*	9.84%*	0.0491*
LightGCN	6.82%*	0.0443*	13.98%*	0.0736*	10.33%*	0.0520*
SGL	7.19%*	0.0475*	14.45%*	0.0766*	10.32%*	0.0539*
CKE	6.86%*	0.0431*	13.75%*	0.0685*	9.01%*	0.0382*
RippleNet	4.22%*	0.0251*	10.58%*	0.0549*	8.58%*	0.0407*
KGCN	5.32%*	0.0338*	11.11%*	0.0569*	8.87%*	0.0431*
KGAT	6.75%*	0.0432*	13.90%*	0.0739*	9.07%*	0.0442*
KGIN	7.12%*	0.0462*	14.36%*	0.0748*	10.44%*	0.0527*
CKAN	6.89%*	0.0441*	13.80%*	0.0726*	9.91%*	0.0499*
MVIN	6.91%*	0.0441*	13.98%*	0.0742*	9.62%*	0.0487*
KGCL	**7.56%**	**0.0493**	**14.96%**	**0.0793**	**10.73%**	**0.0551**

- **KGAT** [39]. This model designs an attentive message passing scheme over the knowledge-aware collaborative graph for embedding fusion. The relevance of neighboring nodes are differentiated during the propagation process.
- **KGIN** [40]. It is a recently proposed KG-enhanced recommendation model to identify latent intention of users, and further performs the relational path-aware aggregation for both user-intent-item and KG triplets.
- **CKAN** [42]. It introduces a heterogeneous propagation mechanism to determine the importance of knowledge-aware neighbors, so as to integrate the collaborative filtering representation space with the knowledge graph embedding.
- **MVIN** [29]. It is a multi-view item embedding network based on graph neural architecture. Information from both user and entity side is considered to learn feature embeddings of items.

Parameter Settings. Our proposed KGCL is implemented with PyTorch. Most of compared baselines are evaluated based on the unified recommendation library *RecBole* [54]. In particular, we fix the embedding dimensionality as 64 for all methods, and conduct the model optimization with the learning rate of $1e^{-3}$ and batch size of 2048. For knowledge-aware recommendation models, the number of context hops and memory size is set as 2 and 8, respectively. In our KGCL, we search the temperature parameter τ and contrastive loss balance parameter λ_1 in the range of {0.1,...,0.5,...,1.0} with an increment of 0.1. Additionally, truncation probability p_τ and p_a are searched among the range of $\{0.6, 0.7, 0.8, 0.9\}$.

4.2 Performance Comparison with SOTA (RQ1)

We report the overall performance evaluation of all methods in Table 2. From the results, we summarize the following observation:

- KGCL consistently performs better than other baselines in all cases, which verifies the effectiveness of integrating knowledge graph embedding into the contrastive learning paradigm. The diversity of evaluation datasets varying by sparsity degrees, knowledge graph characteristics, and recommendation scenarios. The superior results justify the generality and flexibility of our KGCL framework. Overall, the improvements obtained by KGCL can be attributed to two aspects: i) Benefiting from our knowledge graph contrastive learning, KGCL can denoise entity-dependent relationships and capture accurate item-wise semantics. ii) KGCL is able to guide the interaction data augmentation schema for self-supervised information with the distilled item knowledge.
- We can observe that most of knowledge-aware recommender systems achieve better performance as compared to BPR and NCF. This confirms the helpfulness of incorporating knowledge graph information to tackle the sparsity issue in collaborative filtering. Among various knowledge-aware methods, KGIN performs the best by enhancing user representation with the exploration of the latent intention, based on the intent-aware relational paths for embedding propagation. The performance gap between our KGCL and other knowledge-aware models (*e.g.,* KGAT, CKAN, MVIN), suggests that the noisy knowledge graph misleads the learning of item-item semantic relatedness.
- The relatively superior performance achieved by SGL indicates the rationality of generating self-supervised signals from unlabeled user behaviors, to improve the robustness of recommendation. Different from the self-supervised recommendation model SGL, our KGCL creates contrastive self-supervision signals with knowledge-guided augmentation schema, which effectively incorporates the KG-based item semantic relatedness to alleviate the interaction sparsity issue in a robust and explicit manner.

Table 3: Impact study of knowledge-aware graph augmentation schema with model variants of KGCL.

Model	Amazon-Book		MIND	
	Recall	NDCG	Recall	NDCG
KGCL	**14.96%**	**0.0793**	**10.73%**	**0.0551**
KGCL w/o KGA	14.85%	0.0788	10.57%	0.0546
KGCL w/o KGC	14.68%	0.0771	10.35%	0.0537

Table 4: Impact of τ and λ_1 on Amazon-Book dataset.

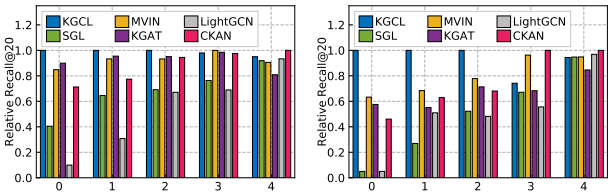
Metric	Recall@20				
λ_1, τ	0.1	0.2	0.3	0.4	0.5
10^{-1}	12.93%	**14.96%**	14.46%	13.94%	13.17%
10^{-2}	13.74%	13.68%	13.08%	12.39%	11.55%
10^{-3}	12.77%	11.94%	11.27%	10.62%	9.97%

4.3 Ablation Study of KGCL Framework (RQ2)

Impact of Knowledge-aware Graph Augmentation Schema. We investigate the effect of our knowledge-aware graph augmentation schema from the views of both knowledge graph and user-item interaction behaviors. accordingly, we design two model variants:

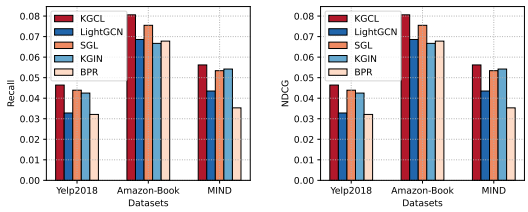
- i) "w/o KGA": the variant of KGCL without the knowledge-guided augmentation scheme on user-item interaction graph. Instead, the contrastive views of interaction graph are constructed with randomly edge sampling, for mutual information estimation.
- ii) "w/o KGC": we remove the knowledge graph contrastive learning component from KGCL, and directly forward the item representations encoded from our relation-aware knowledge aggregator into the graph-based CF framework for contrastive learning.

From results in Table 3, it is clear that the performance superiority of our KGCL framework can be achieved in all cases. This fact indicates that the proposed knowledge-guided contrastive learning over the interaction graph, and the knowledge graph contrastive



(a) Amazon-Book (b) Yelp2018

Figure 5: Performance with different interaction density degrees of items between KGCL and baselines. Recall values are normalized to range [0, 1] for better presentation.



(a) Recall (b) NDCG

Figure 6: Comparison on cold-start users.

Table 5: Performance in alleviating KG noise.

Model	Yelp2018		Amazon-book		MIND		**Avg. Dec.**
	Recall	NDCG	Recall	NDCG	Recall	NDCG	
KGAT	6.51%	0.0409	13.29%	0.0639	8.73%	0.0370	13.57%
KGIN	6.85%	0.0444	13.69%	0.0719	10.32%	0.0527	3.37%
MVIN	6.65%	0.0416	13.28%	0.0703	9.31%	0.0424	8.81%
KGCL	**7.52%**	**0.0490**	**14.93%**	**0.0787**	**10.69%**	**0.0550**	**0.58%**

learning are both effective for making better recommendations.

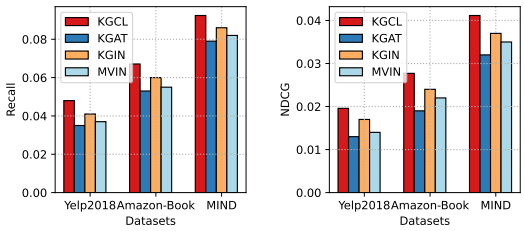
Hyper-parameter Sensitivity. We further present the evaluation results of our hyperparameters of λ_1 and τ for controlling the strength of contrastive regularization and hard negative sampling, respectively. In particular, λ_1 and τ are searched from the range of (10^{-1}, 10^{-2}, 10^{-3}) and (0.1, 0.2, 0.3, 0.4, 0.5), respectively. We can observe that the best performance can be achieved by $\lambda_1 = 0.1$ and $\tau = 0.2$, which indicates that larger value of τ may limits the discrimination ability between different negative instances. In addition, small value of λ_1 corresponds to the less influence of contrastive optimization loss on the main embedding space.

4.4 Benefits of KGCL in Alleviating Data Sparsity and Noise Effect (RQ3)

In this subsection, we investigate the robustness of our KGCL by evaluating its performance for handling sparse and noisy data.

Sparse User Interactions. To investigate the robustness of our KGCL in handing users without sufficient interactions, we follow similar settings in [50] to generate sparse user set with less than 20 interactions for Yelp2018 and Amazon-Book, and 5 interactions for MIND data. The results on sparse users are reported in Figure 6.

Long-tail Item Recommendation. To justify the effect of our KGCL in long-tail item recommendation, we split all items into five groups with equal number of items (interaction density increases from group 0 to group 4). Separated evaluations are conducted on different item groups. Results are shown in Figure 5.



(a) Recall (b) NDCG

Figure 7: Recommendation performance comparison on items connected to long-tail KG entities.

Knowledge Graph Noise. We investigate the robustness of our KGCL by separately injecting noisy triplets to the knowledge graph and testing on items connected with long-tail entities in the KG. In particular, we firstly randomly add 10% noisy triplets into the existing KG data with the unchanged test set, to simulate the situation where the collected KG has a large number of topic-irrelevant entities. In addition, to simulate the KG noise scenario caused by long-tail entities, we collect the 20% long-tail entities and filter the items connected with these long-tail entities in the testing data to perform evaluations. The results are reported in Table 5 and Fig 7.

- KGCL consistently outperforms SGL in all cases, which justifies the superiority of our knowledge graph-guided contrastive learning as compared to the randomly dropout-based strategy. Furthermore, the performance improvement of our KGCL over KGIN, indicates the necessity of incorporating knowledge graph-guided self-supervised signals into knowledge-aware recommender systems, so as to address the issue of noisy knowledge graph in misleading the encoding of user preference for recommendation.
- Considering the recommendation scenario in which items often exhibit long-tail distribution, our KGCL significantly improves the recommendation performance for long-tail items. This observation again demonstrates the superiority of our KGCL method in alleviating popularity bias for recommendation. However, unpopular items are less likely to be recommended by other baselines. Moreover, the performance superiority of our KGCL compared with competitive KG-enhanced recommender systems (*e.g.*,, KGAT, CKAN), indicates that blindly incorporating knowledge graph information into collaborative filtering may involve item relation noise, and cannot effectively alleviate popularity bias.
- Our KGCL always achieves the best performance when competing with state-of-the-art knowledge-aware recommendation models, in distilling useful information from noisy knowledge graph to assist the modeling of user preference. Specifically, KGCLachieves lowest average performance decreasing in alleviating KG noise (Table 5), and the best evaluation results on items with sparse knowledge entities (Figure 7). This verifies the rationality of our knowledge graph contrastive learning paradigm in discovering relevant item semantics from noisy KG information.

4.5 Case Studies (RQ4)

We perform case studies with sampled examples for news recommendation, to show the inference results with and without our knowledge graph contrastive learning (as illustrated in Figure 8).

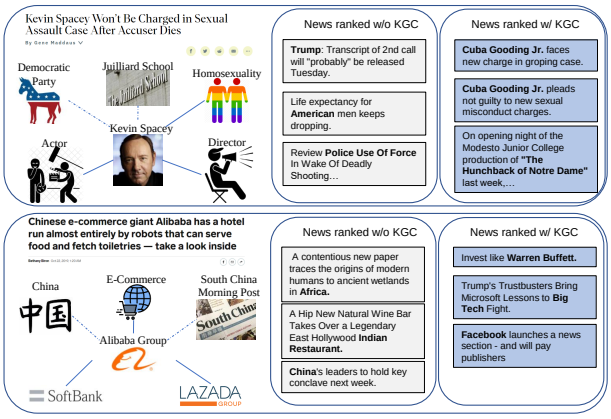


Figure 8: Two examples of relevant news ranked w/ and w/o KGCL. News in blue color means semantically relevant and bold font means the entities extracted from the news item.

The first news case is about a celebrity *Kevin Spacey* and his controversial affairs. We show entities connected with this news in the provided KG information, where his political position *Democratic Party* and his graduated school *Julliard School* are obviously irrelevant to this news. These noisy KG information may mislead the user representation by introducing biased information about politics or education. On the right part, we show three most similar news with this sampled news, ranked by the model with and without the KG-aware contrastive learning, respectively. Specifically, we can see that similar news ranked by model without KGC all relate to the theme of nationality and politics. Among them, the news topics are about *Donald Trump*, *American People* and *Police Use of Force*, which are all irrelevant to the movie celebrity *Kevin Spacey*. In contrast, the results inferred by our KGCL are all closely relevant to this news, *i.e.,* movie celebrity *Cuba Gooding Jr.* or the movie *The Hunchback of Notre Dame*. Coincidentally, *Cuba Gooding Jr.* faces similar legal charges with *Kevin Spacey*. After effectively denoising KG information, our method can correlate *Cuba Gooding Jr.* and *Kevin Spacey* with each other for accurate recommendation.

Another news example about China's tech giant *Alibaba* is shown in the below part. Analogously, observed noisy entities *China* and *South China Morning Post* may influence the learning process of item-wise semantic relatedness towards the media press and nationality news. We can observe that the recommended similar news are all about national news from *China*, *India* and *Africa*. By integrating our knowledge graph contrastive learning component, our model allows the recommendation framework to capture accurate semantic dependency among items by debiasing noisy entity-dependent relationships. In particular, the news identified by our KGCL is specific to the big tech companies, which are very relevant to the target news. Overall, KGCL is able to deconfound recommender system for alleviating knowledge graph information bias and eliminating the impact of irrelevant entities.

5 RELATED WORK

Knowledge Graph-enhanced Recommendation. Prior methods of KG-enhanced recommendation methods can be roughly grouped into two categories: embedding-based methods and path-based methods. For embedding-based methods [30, 36, 48, 51], they leverage relations and entities in the KG to enhance the semantic representations in recommender systems. Usually, these methods apply a transition constrain to learn meaningful knowledge embedding for users and items. For example, CKE [51] incorporates different types of side information into the collaborative filtering framework. In CKE model, the embedding of items' structural knowledge is encoded with TransR [19], and the textual and visual knowledge are learned with the proposed auto-encoder. Another representative method is DKN [36], which integrates the semantic representations of news to learn better item embeddings.

Path-based methods [13, 35, 37, 39, 45] aim to explore the potential information between items in knowledge graphs by constructing meta-path for information propagation. For instance, MCRec [13] designs meta-path-based mutual attention mechanism for Top-N recommendation, which produces user, item, and meta-path-based context representations. Overall, they offer relatively superior performance as compared to most embedding-based methods, since high-order knowledge-aware dependencies can be captured in those approaches. However, path-based methods highly depend on the design of meta-paths, which relies on the domain knowledge and human efforts. Moreover, aggregating information along different meta-paths is very time-consuming, resulting in inefficient knowledge-aware recommender systems.

Contrastive Learning for Recommender System. Recently, contrastive learning has attracted much attention in offering self-supervised signals for various domains, nature language processing [8] and image data analysis [6]. It aims to learn quality discriminative representations by contrasting positive and negative samples from different views. Several recent attempts have brought the self-supervised learning to the recommendation [21, 22, 43, 44]. For example, SGL [44] performs dropout operations over the graph connection structures with different strategies, *i.e.,*, node dropout, edge dropout and random walk. Additionally, CML [43] enhances the recommender system with the consideration of multi-behavior relationships between users and items with contrastive learning. Motivated by these existing contrastive learning frameworks, this work develops a new graph contrastive learning paradigm for recommendation by effectively integrating knowledge graph representation and user-item interaction augmentation.

6 CONCLUSION

In this work, our proposed KGCL framework performs the initial attempts to explore the knowledge graph semantics and alleviate the data noise issue for recommendation under a knowledge-guided contrastive learning paradigm. The KG-aware data augmentation is conducted to investigate auxiliary self-supervised signals, based on estimating the effect of knowledge ambiguous items for user preference learning. This work opens up new research possibilities for knowledge-aware recommender systems. Extensive experiments on several real-world datasets have demonstrated the superiority of KGCL as compared to various state-of-the-art methods.

ACKNOWLEDGMENTS

This research is supported by the research grants from the Department of Computer Science & Musketeers Foundation Institute of Data Science at the University of Hong Kong.

REFERENCES

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

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *NIPS* 26 (2013).

[3] 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.

[4] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*. 335–344.

[5] Ting Chen, Simon Kornblith, et al. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. PMLR, 1597–1607.

[6] Yu Deng, Jiaolong Yang, et al. 2020. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *CVPR*. 5154–5163.

[7] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *WWW*. 417–426.

[8] Hao Fu, Shaojun Zhou, Qihong Yang, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. LRC-BERT: Latent-representation Contrastive Knowledge Distillation for Natural Language Understanding. In *AAAI*, Vol. 35. 12830–12838.

[9] Zhabiz Gharibshah and Xingquan Zhu. 2021. User Response Prediction in Online Advertising. *ACM Computing Surveys (CSUR)* 54, 3 (2021), 1–43.

[10] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. 355–364.

[11] 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.

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[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 *KDD*. 1531–1540.

[14] Chao Huang. 2021. Recent Advances in Heterogeneous Relation Learning for Recommendation. *arXiv preprint arXiv:2110.03455* (2021).

[15] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xiao, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. 2021. Knowledge-aware coupled graph neural network for social recommendation. In *AAAI*.

[16] Weike Jin, Zhou Zhao, Pengcheng Zhang, Jieming Zhu, Xiuqiang He, and Yueting Zhuang. 2021. Hierarchical Cross-Modal Graph Consistency Learning for Video-Text Retrieval. In *SIGIR*. 1114–1124.

[17] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, et al. 2020. Supervised contrastive learning. In *NIPS*.

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

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

[20] Shang Liu, Zhenzhong Chen, et al. 2019. User-video co-attention network for personalized micro-video recommendation. In *WWW*. 3020–3026.

[21] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive Self-supervised Sequential Recommendation with Robust Augmentation. *arXiv preprint arXiv:2108.06479* (2021).

[22] Xiaoling Long, Chao Huang, Yong Xu, Huance Xu, Peng Dai, Lianghao Xia, and Liefeng Bo. 2021. Social Recommendation with Self-Supervised Metagraph Informax Network. In *CIKM*. 1160–1169.

[23] 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.

[24] Jay Pujara, Eriq Augustine, and Lise Getoor. 2017. Sparsity and noise: Where knowledge graph embeddings fall short. In *ACL*. 1751–1756.

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

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

[27] Dana Ruiter, Cristina Espana-Bonet, and Josef van Genabith. 2019. Self-supervised neural machine translation. In *ACL*. 1828–1834.

[28] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *WWW*. 111–112.

[29] Chang-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.

[30] Yu Tian, Yuhao Yang, Xudong Ren, Pengfei Wang, Fangzhao Wu, Qian Wang, and Chenliang Li. 2021. Joint Knowledge Pruning and Recurrent Graph Convolution for News Recommendation. In *SIGIR*. 51–60.

[31] Riku Togashi, Mayu Otani, and Shin'ichi Satoh. 2021. Alleviating Cold-Start Problems in Recommendation through Pseudo-Labelling over Knowledge Graph. In *WSDM*. 931–939.

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

[33] Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. 2021. Towards domain-agnostic contrastive learning. In *ICML*. PMLR, 10530–10541.

[34] 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.

[35] 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.

[36] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*. 1835–1844.

[37] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge Graph Convolutional Networks for Recommender Systems. In *WWW*. 3307–3313.

[38] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. 2020. Time to Shop for Valentine's Day: Shopping Occasions and Sequential Recommendation in E-commerce. In *WSDM*. 645–653.

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

[40] 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.

[41] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*, Vol. 33. 5329–5336.

[42] 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.

[43] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive Meta Learning with Behavior Multiplicity for Recommendation. In *WSDM*. 1120–1128.

[44] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, et al. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[45] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. 2021. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In *AAAI*, Vol. 35. 4486–4493.

[46] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. 2021. Graph meta network for multi-behavior recommendation. In *SIGIR*. 757–766.

[47] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, et al. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *SIGIR*. 285–294.

[48] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. 2019. Relational collaborative filtering: Modeling multiple item relations for recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 125–134.

[49] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, et al. 2020. Graph contrastive learning with augmentations. *NIPS* 33 (2020), 5812–5823.

[50] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *WWW*. 413–424.

[51] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, et al. 2016. Collaborative knowledge base embedding for recommender systems. In *KDD*. 353–362.

[52] Huan Zhao, Quanming Yao, Jianda Li, et al. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In *KDD*. 635–644.

[53] Wayne Xin Zhao, Gaole He, Kunlin Yang, Hongjian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. Kb4rec: A data set for linking knowledge bases with recommender systems. *Data Intelligence* 1, 2 (2019), 121–136.

[54] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, et al. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM*.

[55] Chenyi Zhuang and Qiang Ma. 2018. Dual graph convolutional networks for graph-based semi-supervised classification. In *WWW*. 499–508.