Journal of Intelligent Information Systems
https://doi.org/10.1007/s10844-023-00778-0

Bi-knowledge views recommendation based on user-oriented contrastive learning

Yi Liu[1] · Hongrui Xuan[1] · Bohan Li[1,2,3]

Received: 1 October 2022 / Revised: 19 January 2023 / Accepted: 19 January 2023

Abstract

In recommender system, knowledge graph (KG) is usually leveraged as side information to enhance representation ability, and has been proven to mitigate the cold-start and data sparsity issues. However, due to the complexity of KG construction, it inevitably brings a large amount of noise, thus simply introducing KG into recommender system may hurt the performance of models. In addition, the current KG-based recommendation models mainly include the following issues: (1) The rich facts and semantic knowledge contained in KG are not fully explored. (2) The useless noise in KG is not effectively filtered, and the representation obtained by neighborhood aggregation shows poor quality. (3) Nodes with long-tail distribution are easily ignored and the models fail to balance the attention between popular and unpopular items. Therefore, we propose a Bi-Knowledge Views Recommendation Based on User-Oriented Contrastive Learning architecture (BUCL) to improve the representation quality and alleviate the long-tail distribution of entities. In particular, different graph embedding methods are applied to fully extract the rich facts and semantic knowledge in the KG to obtain multiple views of nodes. Based on the different representation views, a user-oriented item quality estimation method is proposed to guide the model to generate multiple augmented subgraphs. Each node provides enough negative samples to ensure that the model discriminates the same node from other nodes in differentiated subgraphs with contrastive learning. Experiments on three benchmark datasets show that BUCL consistently outperforms state-of-the-art models, alleviating the long-tail distribution problem and reducing the impact of noise.

Keywords Recommender system · Contrastive learning · Knowledge graph · Data augmented

✉ Bohan Li
bhli@nuaa.edu.cn

1 College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China

2 Ministry Key Laboratory for Safety-Critical Software Development and Verification, Nanjing University of Aeronautics and Astronautics, Nanjing, China

3 National Engineering Laboratory for Integrated Aero-Space-GroundOcean Big Data Application Technology, Xi'an, China

Published online: 13 April 2023

1 Introduction

With the rapid development of the Internet and information computing, we have entered an era of information overload. Massive data are generated all the time, but the amount of information is beyond what an individual can accept, process, or use effectively (Desrosiers & Karypis, 2011). As an information filtering system (Gao et al., 2022), recommender system can explore users' preferences based on their historical interactions, help users find the content they pursue from the massive information, and improve the efficiency of information distribution and acquisition. Due to its effectiveness, recommender system has been successfully applied to news recommendation (Ge et al., 2020), e-commerce recommendation (Wang et al., 2018c), music recommendation, etc (Atas et al., 2021; Dai et al., 2022; Lei et al., 2022).

Collaborative filtering (CF) is a widely used and efficient algorithm to recommend items to users by mining users' historical interactions, which occupies an extremely important position in the recommendation field (Koren et al., 2022). It applies the matrix factorization (MF) method to decompose the user-item interaction matrix and then capture their relations in the embedded space. However, CF methods have limitations in processing complex data input, and also face the problems of data sparsity and cold-start (Panda & Ray, 2022).

To solve the above problems, researchers try to introduce knowledge graph (KG) as side information to enrich the connecting edges of items. KG is a structured semantic knowledge base composed of triples, which represents entities and their relations in the physical world in a graph structure (Yan et al., 2020). Since it contains rich facts and fruitful semantic knowledge (Wang et al., 2017), and can alleviate the problems of data sparsity and cold-start, KG-based recommendation has attracted great attention from researchers. In KG, triples are in the form of (head, relation, tail), recording facts and semantic information. When KG is introduced into recommender system, the nodes in KG are usually associated with items as entities, and item representation is obtained by extracting the interaction relations through neighborhood aggregation. Although effective, KG-based recommendation is built on a robust KG, and previous work did not consider noise filtering from the users' perspectives. As shown in Fig. 1, users u_1 and u_2 have watched 'Inception', which has the entities 'Leonardo DiCaprio' and 'Science fiction'. Since all the other films u_2 watches have the entity 'Leonardo DiCaprio', we speculate that actor is the reason why u_2 watches 'Inception'. Correspondingly, the reason why u_1 watches 'Inception' may be because the film genre is 'Science fiction'. Therefore, we need to further consider the perspective bias caused by users' attention to build differentiated views of users. Moreover, items in KG have a significant long-tail distribution, that is, most items have only a few connections. This phenomenon causes unpopular items to be ignored by the weak generalization model, resulting in lower recommendation accuracy.

Inspired by previous work (Wu et al., 2021; Yang et al., 2022; Jiao et al., 2020; Yu et al., 2022), we adopt the method of contrastive learning to alleviate the aforementioned issues. In recent years, contrastive learning has been widely used in the fields of CV (Cole et al., 2022; He et al., 2020a) and NLP (Wu et al., 2020b), and the application of recommender system is relatively small, but its effect cannot be denied. Contrastive learning is an implementation of self-supervised learning (SSL) (Wu et al., 2021; Zou et al., 2022). Essentially, it constructs positive and negative samples for comparison through different data augmented methods, and as an auxiliary learning task to jointly train with the other SSL tasks. In this paper, we study contrastive learning as an auxiliary task to solve the problems of noise interaction and long-tail distribution in recommender system. To this end, we propose a new architecture **B**i-Knowledge Views Recommendation Based on **U**ser-Oriented **C**ontrastive **L**earning (BUCL) which contains three main components: (1) bi-knowledge views; (2) user-oriented item quality estimation; (3) multi-task training. Firstly, different graph embedding methods

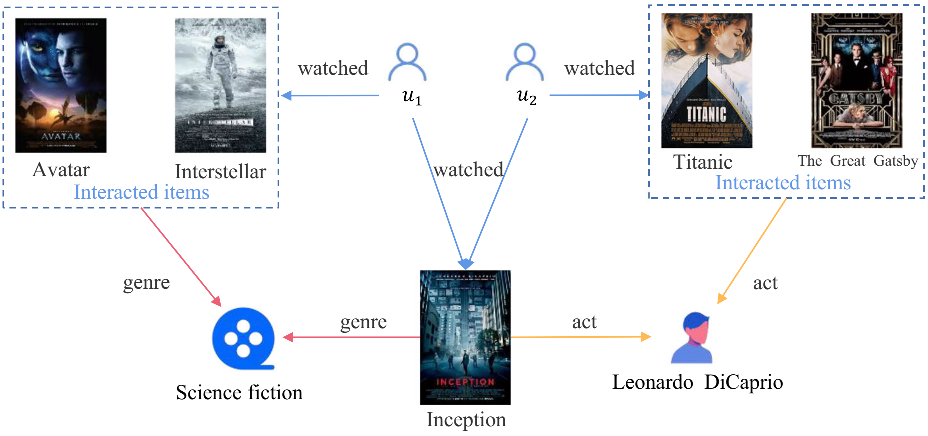

Fig. 1 Example of KG-based recommendation and user perspective bias

TransR (Lin et al., 2015) and TATEC (García-Durán et al., 2014) are applied to focus on the factual and semantic information of triples in the KG respectively to obtain different KG views. These two views have the same graph structure but different refined representations, providing a solid platform for subsequent operations. To construct contrast augmented subgraphs, a user-oriented item quality estimation method based on refined representations is proposed to obtain item quality score. This score takes users' preferences into account to assess item quality and as a signal guide the construction of augmented subgraphs. After that, we adopt InfoNCE (Chen et al., 2020) loss to learn the representation of augmented subgraphs to maximize the similarity of the positive and minimize the similarity of the negative, thus helping the model to correctly distinguish the same instance from others.

BUCL applies a multi-task training strategy that integrates main recommendation loss and contrastive loss since they share similar targets. Through comparative and ablation experiments, BUCL achieved remarkable improvement in the top-k recommendation task, and confirmed the effectiveness of each key component. Overall, the main contributions of this paper are as follows:

- We analyze the rich knowledge contained in KG, and adopt varied graph embedding methods to fully extract factual and semantic information to generate KG subgraphs with different focus points.
- We propose a user-oriented item quality estimation method, which calculates the item quality score from the users' perspectives and then guides the construction of contrast subgraphs with less noise interaction.
- We introduce the InfoNCE (Chen et al., 2020) loss into the KG-based recommender system as an auxiliary learning task to regularize and benefit the recommendation task, alleviate the impact of long-tail distribution, and improve the performance of the model.
- Experimental results on three benchmark datasets show that our model can achieve significant improvement compared with state-of-the-art recommendation methods.

2 Related work

2.1 KG-based recommendation

The triples in the KG contain a large amount of factual and semantic information, which entities are associated with items to enrich their features. There are usually three types:

(1) embedding-based (Zhang et al., 2016; Wang et al., 2018b; Cao et al., 2019) methods directly leverage entities and relations in KG by graph embedding method, and mining semantic information in KG to enrich the representations of users and items. For example, CKE (Zhang et al., 2016) encodes the structure knowledge and context knowledge of items by TransR, and integrates multiple side information into MF (matrix factorization). (2) Path-based methods (Hu et al., 2018; Shi et al., 2018; Wang et al., 2019d; Xian et al., 2019) define or automatically construct meta-paths in advance, and then mine entity connection information. Due to clear recommendation reasons, these methods usually have better explainability. For example, under the restriction of sequence length, RKGE (Sun et al., 2018) automatically constructs the path from the user to the item and inputs it into RNN (recurrent neural network) (Zaremba et al., 2014) to capture entity connection. (3) GNN-based methods combine the above two, and obtain the semantic and connection information between entities. The multi-hop neighbor information of entities is obtained by neighborhood aggregation, which enriches entity representation and realizes more accurate recommendation. For example, KGAT (Wang et al., 2019b) uses TransR to obtain the semantic representations of entities in KG, and captures the connection information by the multi-hop propagation method.

2.2 Contrastive learning

Contrastive methods Contrastive learning is a major branch of self-supervised learning (SSL) (Wu et al., 2021; Zou et al., 2022) and has attracted great attention from researchers in recommendation scenarios in recent years. The key point of contrastive learning is the data augmented part to obtain multiple views of each instance. The same instances are seen as positive examples and expected to be as close as possible in the vector space, while the differences are negative and expected as far as possible. This method is usually divided into three types: (1) Structure-level (Liu et al., 2021; Zhang et al., 2021; Xie et al., 2021; Wang et al., 2021) contrast method carries out some minor perturbations on the graph structure, which do not change the semantic information. Therefore, self-supervised signals are obtained by comparing different graph structures. For example, SGL (Wu et al., 2021) conducted three graph augmented methods node dropout, edge dropout, and random walk, and obtained two views of user-item bipartite graph through the same but independent augmented method. Then LightGCN (He et al., 2020b) is applied to obtain the node representation of the augmented graphs. (2) Feature-level (Yao et al., 2021; Guo et al., 2022) contrast method obtains feature augmented view by modifying the input features. For example, CFM (Yao et al., 2021) uses a two-tower framework, applying masking and dropout methods to modify item features to obtain augmented views. (3) Model-level (Qiu et al., 2022; Yu et al., 2021) contrast method compares augmented views by modifying the model structure. For example, DuoRec (Qiu et al., 2022) applies two different dropout masks on neurons to perturb the model to obtain the augmented views at the model level.

Contrastive learning in collaborative filtering Collaborative filtering is a fundamental paradigm that projects users and items to latent vector space to find their similarity, but the sparsity of interaction limits the representation ability of collaborative filtering. A series of recent studies have combined collaborative filtering methods with contrastive learning to build augmented views to provide external self-supervised signals for collaborative filtering, thereby improving the representation quality of recommender systems. For example, HCCF (Xia et al., 2022) constructs hypergraph structure and collaborative filtering encoding to jointly capture global and local collaborative relations in a cross-view manner. DCL

(Liu et al., 2021) performs complex edge dropout on the ego-network to obtain neighborhood augmented subgraphs, which provide self-supervised signal by maximizing the representation of the same instance.

Contrastive learning in KG-based recommendation Existing KG-based recommendation methods begin to combine contrastive learning, mining the factual and semantic information in KG to provide a fine-grained augmented view and obtain robust self-supervised signals to improve recommendation results. Moreover, with the rich semantics of the KG, combined with self-supervised signals, refined representations are obtained to further alleviate the problems of cold-start and long-tail distribution. For example, MCCLK (Zou et al., 2022) applies KG to construct structural and semantic views, and links collaborative views to perform contrastive learning in three views, mining comprehensive graph structure features and semantic information in a self-supervised manner. KGCL (Yang et al., 2022) obtains additional supervised signals from KG augmented process to guide the cross-view contrastive learning paradigm while suppressing KG noise during information aggregation.

3 Problem formulation

In the classic recommendation scenario, the set of users is usually defined as $U = \{u_1, u_2, \cdots, u_N\}$, and the set of items is $I = \{i_1, i_2, \cdots, i_M\}$, where N and M represent the number of users and items. The user-item interaction matrix $Y = \{y_{u,i}|u \in U, i \in I\}$ is constructed by the historical behavior between users and items. If there is an observed interaction between user u and item i, then $y_{u,i} = 1$, otherwise $y_{u,i} = 0$. Based on the interaction matrix $Y = \{y_{u,i}|u \in U, i \in I\}$, we build a bipartite graph $\mathcal{G}^u = (V_{u,i}, \mathcal{E}_{u,i})$, where $V_{u,i} = U \cup I$ and $\mathcal{E}_{u,i}$ represents the connection edges between users and items. At the same time, KG is introduced as side information such as item entities to compensate for the data sparsity problem. The triples in the KG are defined as $\{(h, r, t)|h, t \in E, r \in R\}$, where E and R respectively denote the set of entities and relations, and (h, r, t) means there is a relation r from head entity h to tail entity t. Similarly, we define $\mathcal{G}^k = (V_{i,e}, \mathcal{E}_e)$ to represent KG, where $V_{i,e} = I \cup E$ and $\mathcal{E}_e$ represents edges between entities. Now, depending on the given $\mathcal{G}^u$ and $\mathcal{G}^k$, our goal is to predict whether user u is interested in item i that had no interaction before. The prediction score is obtained by constructing the prediction function $\hat{y}_{u,i} = \mathcal{F}(u, i; \Theta, \mathcal{G}^u, \mathcal{G}^k)$, where $\hat{y}_{u,i}$ represents the probability of user u may adopt item i, and Θ is the parameter of the model $\mathcal{F}$.

4 Methodology

Now we propose BUCL, the overall workflow is shown in Fig. 2. Relation-aware attention layer is applied to obtain item representations in the sampled KG. User-oriented item quality estimation module calculates the sampling probability $\tilde{P}_{u,i}$, where v_i' and v_i'' are item representations of different sampling KG in the same group. After sampling $\mathcal{G}^u$ with different probabilities, splicing corresponding KG views to obtain the augmented subgraphs which are used for contrastive learning to calculate the contrastive loss. The BPR loss part is realized by LightGCN. In the following sections, we elaborate on (1) bi-knowledge views, which describe the KG from multiple focus points. (2) user-oriented item quality estimation, emphasizing the importance of users' perspectives differences in constructing augmented

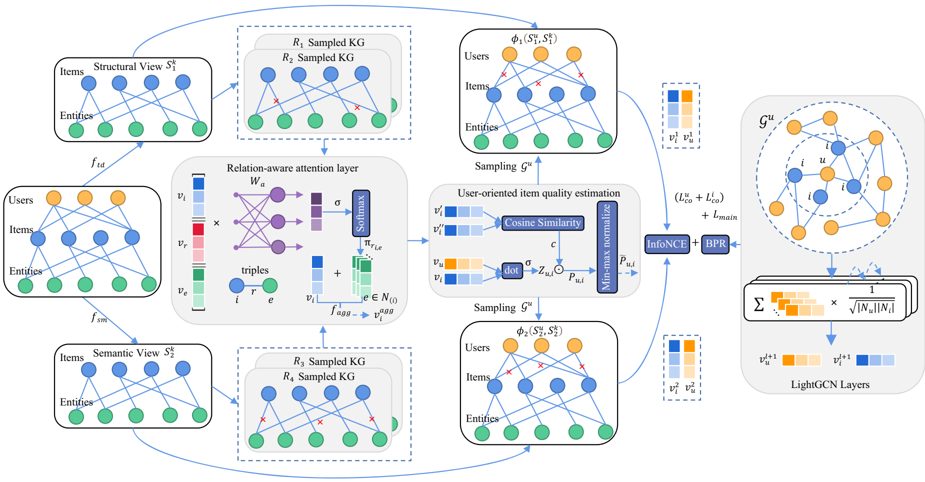

Fig. 2 Overview of the proposed BUCL framework

subgraphs. (3) multi-task training, a joint self-supervised learning paradigm that integrates two losses.

4.1 Bi-knowledge views

To make full use of KG, we divide bi-knowledge views into two parts: graph embedding and relation-aware attention. The former captures various knowledge to generate KG views, and the latter aggregates neighbor features to obtain refined item representation.

4.1.1 Graph embedding

KG contains rich factual and semantic information, and sufficiently extracts the content of KG to benefit subsequent operations. Graph embedding methods project entities and relations into continuous vector space, preserving topology and latent semantics. For that, we exploit the translational distance model TransR (Lin et al., 2015) and the semantic matching model TATEC (García-Durán et al., 2014) to generate two different KG views with different emphases. TransR (Lin et al., 2015) uses the distance scoring function to evaluate the credibility of facts, while TATEC (García-Durán et al., 2014) uses the similarity scoring function to match latent semantics in vector space. And since the two methods have a similar number of parameters and matching computation complexity, the resulting KG views can be treated as equally important. The specific formula is as follows:

$$
\begin{aligned}
f_{td}(\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t) &= -\| M_r^{td} \boldsymbol{v}_h + \boldsymbol{v}_r - M_r^{td} \boldsymbol{v}_t \|_2^2 \\
f_{sm}(\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t) &= \boldsymbol{v}_h^T M_r^{sm} \boldsymbol{v}_t + \boldsymbol{v}_h^T \boldsymbol{v}_r + \boldsymbol{v}_t^T \boldsymbol{v}_r + \boldsymbol{v}_h^T D \boldsymbol{v}_t
\end{aligned} \tag{1}
$$

where $(\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t)$ is the embedding of triples in the KG, $M_r^{td} \in \mathbb{R}^{d \times d}$ and $M_r^{sm} \in \mathbb{R}^{d \times d}$ are projection matrices from the entity space to the relation space of r, and D is a diagonal matrix shared across all different relations. Based on KG, the embedding sets of triples are initialized to obtain KG views $S_1^k = \{(\boldsymbol{v}_h', \boldsymbol{v}_r', \boldsymbol{v}_t')\}$ and $S_2^k = \{(\boldsymbol{v}_h'', \boldsymbol{v}_r'', \boldsymbol{v}_t'')\}$ with the same graph structure but different vector spaces. Then, the loss functions of L^{td} and L^{sm}

corresponding to f_{td} and f_{sm} are constructed for each view and optimized to obtain refined representations. The loss functions are shown as follows:

$$
\begin{aligned}
L^{td} &= \sum_{(\boldsymbol{v}_h,\boldsymbol{v}_r,\boldsymbol{v}_t,\boldsymbol{v}_{t'})\in\mathcal{S}_1^k} -ln\,\sigma\left(f_{td}(\boldsymbol{v}_h,\boldsymbol{v}_r,\boldsymbol{v}_{t'}) - f_{td}(\boldsymbol{v}_h,\boldsymbol{v}_r,\boldsymbol{v}_t)\right)\\
L^{sm} &= \sum_{(\boldsymbol{v}_h,\boldsymbol{v}_r,\boldsymbol{v}_t,\boldsymbol{v}_{t'})\in\mathcal{S}_2^k} -ln\,\sigma\left(f_{sm}(\boldsymbol{v}_h,\boldsymbol{v}_r,\boldsymbol{v}_{t'}) - f_{sm}(\boldsymbol{v}_h,\boldsymbol{v}_r,\boldsymbol{v}_t)\right)
\end{aligned}
\tag{2}
$$

which t' is obtained by negative sampling the triples (h, r, t) in KG.

4.1.2 Relation-aware attention

Given the KG, the neighborhood aggregation method is applied to capture the features of neighbors and integrate them with the current central node representation during the propagation process (Wu et al., 2020a; Zhou et al., 2020). Since the importance of each neighbor is different, we first leverage the attention mechanism to distinguish the importance of different neighbors. The weight of each item to the neighbor node is $\pi_{r_{i,e}}$, where $r_{i,e}$ represents the relation r between item i and entity e. The way to obtain $\pi_{r_{i,e}}$ is as follows:

$$
\pi_{r_{i,e}} = \frac{exp(\sigma(W_a[\boldsymbol{v}_i \| \boldsymbol{v}_r \| \boldsymbol{v}_e] + b))}{\sum_{e'\in N(i)} exp(\sigma(W_a[\boldsymbol{v}_i \| \boldsymbol{v}_r \| \boldsymbol{v}_{e'}] + b))}
\tag{3}
$$

where $N(i)$ are the neighbors of item i based on relation r, σ is the nonlinear activation function such as LeakyRelu. Both $W_a \in \mathbb{R}^{3d}$ and $b \in \mathbb{R}$ are trainable parameters. Respectively, the embeddings of items, relations, and entities are $v_i \in \mathbb{R}^d$, $v_r \in \mathbb{R}^d$, and $v_e \in \mathbb{R}^d$. Each node representation is then updated by aggregating central node and neighbor features based on weight $\pi_{r_{i,e}}$ as follows:

$$
\begin{aligned}
\boldsymbol{v}_i^{agg} &= f_{agg}(\boldsymbol{v}_i, \{\boldsymbol{v}_e | e \in N(i)\})\\
&= \sigma\left(W\left(\boldsymbol{v}_i + \sum_{e\in N(i)} \pi_{r_{i,e}} \boldsymbol{v}_e\right) + b\right)
\end{aligned}
\tag{4}
$$

For simplicity, we adopt the sum aggregation method, and other aggregation methods can also be used.

4.2 User-oriented item quality estimation

The purpose of our estimation of item quality is to construct reasonable subgraphs with less noise, thus reducing the difficulty of representation learning and enhancing the representation ability of contrastive learning. Inspired by the paper (Yang et al., 2022), high quality nodes tend to obtain similar embedding in different KG structures. However, it neglects different users have unique perspectives on different items, resulting in items that users really care about being abandoned and that the model cannot capture users' preferences correctly. For that, we guide subgraph construction by perturbing the KG structure and building a differentiated view for each user.

4.2.1 Item quality estimation

As a kind of side information, KG plays the role of enriching the representations of items, but it also introduces noise that may significantly reduce the performance of the model. Therefore, we seek to build a robust KG that is less affected by noise interactions.

Firstly, we randomly sampled the KG views S_1^k and S_2^k to generate a series of KG subgraphs. Then, for the KG subgraphs, the f_{agg} method in (4) is adopted to aggregate the center item node and its refined neighbor features to obtain the representations of items. Finally, the similarity of the same source items is calculated by cosine similarity function, and the quality score is initially obtained. The equation is as follows:

$$\begin{aligned}\mathbf{c}' &= s\left(f_{agg}\left(V_i, R_1\left(S_1^k\right)\right), f_{agg}\left(V_i, R_2\left(S_1^k\right)\right)\right)\\ \mathbf{c}'' &= s\left(f_{agg}\left(V_i, R_3\left(S_2^k\right)\right), f_{agg}\left(V_i, R_4\left(S_2^k\right)\right)\right)\end{aligned} \tag{5}$$

where R represents random sampling with different seeds for perturbing the structure of views. $s(\cdot)$ is the cosine similarity function used to calculate the quality scores of $\mathbf{c}' \in \mathbb{R}^{1\times M}$ and $\mathbf{c}'' \in \mathbb{R}^{1\times M}$ respectively based on S_1^k and S_2^k.

4.2.2 User-oriented adjustment

The quality score obtained by perturbing the KG structure reflects the insensitivity of the item structure variation in KG. However, this operation ignores encoding the collaborative effects of user-item interactions. At the same time, each user holds a unique perspective on the items, and high quality scores are not a sufficient condition for users to adopt them. Then, the identical quality score fails to consider the bias of perspective, which loses the users' preferences and reduces the valuable signals of the subgraphs. Therefore, we encode collaborative effects to balance the relations between user perspective and quality score, and guide the construction of differentiated perspectives subgraphs. The formula is as follows:

$$\begin{aligned}Z_{u,i} &= \sigma\left(\boldsymbol{v}_u^T \boldsymbol{v}_i\right)\\ P_{u,i} &= Z_{u,i} \odot \mathbf{c}\\ \tilde{P}_{u,i} &= a + \frac{(b-a)}{P_{max} - P_{min}}\left(P_{u,i} - P_{min}\right)\end{aligned} \tag{6}$$

where σ is softmax function, $Z_{u,i}$ represents the level of attention from user u to item i. Now we perform $Z_{u,i}$ to transform $\mathbf{c}$ to obtain item sampling probability based on users' perspectives. $\odot$ is an element-wise multiplication with a broadcast mechanism. $\tilde{P}_{u,i}$ represents the normalized probability that the connected edge between u and i is preserved. The sampling probability is transformed into the interval $[a, b]$ (e.g., $[0.6, 0.9]$) by min-max normalization, ensuring that each edge has the probability to be preserved or abandoned. The larger the $\tilde{P}_{u,i}$ value is, the easier the edge is retained, and vice versa. Through the above operation, we transform $\mathbf{c}'$ and $\mathbf{c}''$ to obtain the sampling probabilities $\tilde{P}'_{u,i}$ and $\tilde{P}''_{u,i}$.

4.3 Multi-task training

Since contrastive learning and recommendation tasks share similar targets, namely, correctly distinguishing positive and negative samples, we adopt a multi-task training strategy to jointly optimize the main recommendation task and the contrastive learning task to assist in improving the performance of the former.

4.3.1 Contrastive loss

According to these two probabilities $\tilde{P}'_{u,i}$ and $\tilde{P}''_{u,i}$, we sample the observable connection edges between user and item to obtain user-item subgraphs $\mathcal{G}_1^u$ and $\mathcal{G}_2^u$. Since $\mathcal{G}_1^u$ and $\mathcal{G}_2^u$ are generated from the original views S_1^k and S_2^k, we correspond them one by one. Respectively, $S_1^u = \{(v'_u, v'_i)\}$ and $S_2^u = \{(v''_u, v''_i)\}$ are defined as the corresponding representations in $\mathcal{G}_1^u$ and $\mathcal{G}_2^u$. Then we construct two augmented subgraphs $\phi_1(S_1^u, S_1^k)$ and $\phi_2(S_2^u, S_2^k)$ that contain the embedding set of users, items and entities to realize the comparison of representations. Toward the KG views, the representations of items are updated by neighborhood aggregation in (4), and capture the knowledge in S^k refined by graph embedding method. The classic recommendation model LightGCN is adopted for the final representation in the user-item subgraphs to achieve the purpose of simplicity and efficiency, and the formula is as follows:

$$\boldsymbol{v}_u^{(l+1)} = \sum_{i \in N_u} \frac{\boldsymbol{v}_i^l}{\sqrt{|N_u||N_i|}} \quad \boldsymbol{v}_i^{(l+1)} = \sum_{i \in N_i} \frac{\boldsymbol{v}_u^l}{\sqrt{|N_i||N_u|}} \tag{7}$$

where $\frac{1}{\sqrt{|N_u||N_i|}}$ is the layer normalization to ensure the smoothness of data feature distribution. After that, We obtain the representations $(\boldsymbol{v}'_u, \boldsymbol{v}'_i)$ and $(\boldsymbol{v}''_u, \boldsymbol{v}''_i)$ of different augmented subgraphs. As mentioned earlier, we controlled a similar number of parameters and the computation complexity of graph embedding methods. Therefore, two subgraphs are regarded as equally important and then calculate the mean value of the embedding of the same instance to obtain the final representation. For the contrastive learning part of the model, we adopt the contrastive loss of infoNCE (Chen et al., 2020) to maximize the similarity of the same instance in different subgraphs and minimize the similarity of others as follows:

$$L_{co}^u = \sum_{x \in U} -log \frac{exp\left(s\left(\boldsymbol{v}_x^1, \boldsymbol{v}_x^2\right)/\tau\right)}{\sum_{y \in U, x \neq y} exp\left(s\left(\boldsymbol{v}_x^1, \boldsymbol{v}_y^2\right)/\tau\right)} \tag{8}$$

where τ is the temperature parameter to control the smoothness of the softmax curve. L_{co}^u represents the contrastive loss of the user side, and the contrastive loss L_{co}^i on the item side is obtained in the same way.

4.3.2 Objective optimization

The final step is to establish an objective function to optimize the parameters and then the model can obtain the correct prediction results. To optimize the main recommendation model, we choose BPR loss to maximize the difference between positive and negative samples, as follows:

$$L_{main} = \sum_{u \in U} \left(\sum_{i: y_{ui}=1} f(y_{ui}, \hat{y}_{ui}) - \sum_{i': y_{ui'}=0} f\left(y_{ui'}, \hat{y}_{ui'}\right) \right) \tag{9}$$

where $\hat{y}_{uv}$ represents the matching score of user u and item i, conducted by inner product $\hat{y}_{uv} = v_u^T v_i$. In the above, we have given the main recommendation task loss L_{main}, as well as the contrastive loss L_{co}^u and L_{co}^i of the user and item side. Now, we need to integrate contrastive losses into the main loss as a multi-task learning process as follows:

$$L = L_{main} + \lambda_1 \left(L_{co}^u + L_{co}^i\right) + \lambda_2 \|\Theta\|_2^2 \tag{10}$$

where Θ is the set of model parameters, λ_1 balances the ratio between main recommendation task and auxiliary task, λ_2 is the hyperparameter to control the weight of L_2 regularization terms.

5 Experiments

To prove the advantage and effectiveness of our framework, we conducted extensive experiments and answered the following questions:

- **RQ1:** How does our BUCL compare with the state-of-the-art recommendation model?
- **RQ2:** What are the benefits of our proposed modules for overall performance?
- **RQ3:** Compared with other baseline models, does BUCL alleviate the problems of long-tail distribution and noise interaction?
- **RQ4:** What are the effects of different hyperparameter settings on the model?

5.1 Experimental settings

5.1.1 Datasets

We evaluate the performance of our model based on three benchmark datasets, Yelp2018, Amazon-Book, and MIND. These datasets are commonly adopted for recommendation tasks and are available in open source. Table 1 shows the statistics of each dataset and the detailed introduction is as follows:

- **Yelp2018** is a dataset of merchants (restaurants, bars, etc.) recommendations from the 2018 Yelp Challenge. It contains 45,919 users' scores on all kinds of businesses, which are regarded as items. The corresponding KG contains 47,472 entities, 869,603 KG triples, and 42 relations.
- **Amazon-book** is the dataset of book reviews, which comes from Amazon E-commerce platform and is commonly used in KG-based recommendation. It contains 70,679 user ratings of books, and books are regarded as items. The corresponding KG contains 29,714 entities, 686,516 KG triples, and 39 relations.
- **MIND** is a news recommendation dataset established by the users' click behaviors of Microsoft News, which contains 300,000 users and the corresponding KG contains 106,500 entities, 746,270 KG triples, and 90 relations.

Table 1 Statistics of datasets

	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
#Relations	42	39	90
#Entities	47,472	29,714	106,500
#KG triples	869,603	686,516	746,270

5.1.2 Baselines

We demonstrate the advantages of our model by comparing it with other baseline models of different categories. The details of these models are as follows:
Collaborative Filtering Methods

- **BPR** (Rendle et al., 2014) is a framework that leverages user behavior data to obtain personalized ranking of item ratings.
- **NGCF** (Wang et al., 2019c) adopts the neighbor aggregation method to encoder collaborative effects and captures high-order connectivity relations through Graph Convolution Network (GCN).
- **LightGCN** (He et al., 2020b) simplifies the GCN by removing nonlinear activation, and only contains necessary components to achieve lightweight model design and obtain higher training efficiency.

Embedding-based Methods

- **CKE** (Zhang et al., 2016) applies TransR (Lin et al., 2015) and auto-encoders architecture to unify various types of side information (structure and semantic knowledge) to obtain item representation.

Path-based Methods

- **MCRec** (Hu et al., 2018) defines the meta-paths in advance and enters the sampled path instance into CNN (Chen, 2015) to get the path embedding representations after max-pooling operation.

GNN-based Methods

- **RippleNet** (Wang et al., 2018a) combines embedding-based methods and path-based methods to refine users' representation through preference propagation on the KG.
- **KGAT** (Wang et al., 2019b) initializes the KG vector space through TransR, and links users and items with multi-hop neighbor entities through propagation. At the same time, an attention mechanism is used to distinguish the importance of neighbor nodes for aggregation.
- **KGCN** (Wang et al., 2019a) iteratively samples neighbor nodes of fixed numbers and enriches item representation through propagating entities on the KG.
- **CKAN** (Wang et al., 2020) introduces a heterogeneous propagation mechanism to determine the importance of knowledge-aware neighbors, so as to integrate the collaborative filtering vector space with the KG embedding.
- **MVIN** (Zou et al., 2022) proposes that different users have their own perspectives on different items, and then generate a unique item view for each user and realize personalized neighbor aggregation.

Contrastive Learning Methods

- **SGL** (Wu et al., 2021) conducted three graph augmented methods to change the graph structure and then generate multiple views of nodes for contrastive learning.
- **KGCL** (Yang et al., 2022) offers cross-view supervision signals with augmented views to guide the classic recommendation task to obtain robust node representation.

Table 2 Recall and NDCG performance comparisons on Yelp, Amazon-Book, and MIND datasets

model \ dataset	Yelp2018		Amazon-Book		MIND	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR	0.0527	0.0369	0.1375	0.0694	0.0831	0.0379
NGCF	0.0663	0.0412	0.1369	0.0701	0.0882	0.0425
LightGCN	0.0714	0.0452	0.1422	0.0752	0.0909	0.0449
CKE	0.0671	0.0431	0.1395	0.0711	0.0826	0.0361
MCRec	0.0579	0.0391	0.1279	0.0652	0.0796	0.0349
RippleNet	0.0653	0.0398	0.1355	0.0686	0.0843	0.0379
KGAT	0.0727	0.0459	0.1469	0.0757	0.0885	0.0435
KGCN	0.0679	0.0419	0.1386	0.0718	0.0875	0.0411
CKAN	0.0758	0.0475	0.1471	0.0771	0.0912	0.0442
MVIN	0.0741	0.0477	0.1459	0.0767	0.0903	0.0451
SGL	0.0766	0.0489	0.1492	0.0803	0.0945	0.0493
KGCL	0.0772	0.0502	0.1523	0.0805	0.0930	0.0471
BUCL	**0.0801**	**0.0523**	**0.1561**	**0.0841**	**0.0973**	**0.0512**

Bold represents the highest result of all methods

5.1.3 Experiment setup

To evaluate the top-k results of the model, we apply recall@K and ndcg@K metrics which are commonly used in the recommendation scenarios, and set $k = 20$. In other methods, the batch size is fixed to 2048, the learning rate is 3e-4, and the embedding size d is 64. The training parameters are all initialized by Xavier initialization method (Glorot & Bengio, 2010) and using Adam optimizer (Kingma & Ba, 2014) to adapt to the learning rate during the training process. For KG-based methods, we set the number of hops to 3 and the sampling size to 8. In particular, grid search is applied in BUCL framework to fine-tune the hyperparameters. The learning rate is searched in $\{1e^{-3}, 5e^{-4}, 1e^{-4}\}$, the L_2 regularization coefficient λ_2 in $\{1e^{-1}, 1e^{-2}, \cdots, 1e^{-5}\}$, the contrastive loss coefficient λ_1 in $\{2e^{-1}, 2e^{-2}, \cdots, 2e^{-5}\}$, the temperature τ in $\{0.1, 0.2, \cdots, 0.5\}$ and the lower bound of interval a in $\{0.4, 0.5, 0.6, 0.7\}$.

5.2 Overall performance comparison (RQ1)

The comparison results are shown in Table 2, from which we summarized the following aspects:

- BUCL performs better in all datasets, outperforming the other baseline models, proving the superiority and effectiveness of our model. More specifically, it achieved remarkable results with the improvement of the highest baseline w.r.t. NDCG@20 of 4.18%, 4.47% and 3.85% in Yelp2018, Amazon-Book, and MIND, respectively, which verified BUCL is applicable to various scenarios. In summary, through different graph embedding methods, BUCL can fully capture and minging various types of information contained in KG. Moreover, in contrast to KGCL fixed item score, we apply item quality estimation method based on users' perspectives to consider the user perspective bias. By modeling user preferences, the robust augmented subgraphs are constructed from the users' perspectives to reduce the noise and improve the effectiveness of the contrastive learning process. At the same time, contrastive learning improves the exposure of unpopular items and alleviates the problem of long-tail distribution, which achieve better results than other models.

- Through our observation, KG-based models are not necessarily better than CF methods in top-k recommendation. For example, LightGCN performs better than RippleNet and KGCN on all datasets, slightly lower than KGAT on Yelp2018, but higher on MIND dataset. This phenomenon indicates that simply introducing KG as side information may greatly damage the model performance. We put this down to two reasons: (1) Most KG-based methods fail to sufficiently capture the effective information in KG, resulting in the loss of a large number of valuable facts and then hard to perform representation learning. (2) In practice, not all the information in the KG is valuable. Since the noise is not effectively filtered, the robustness of KG cannot be guaranteed, which leads to the failure of the model to fully trust the quality of KG and give appropriate recommendation results.
- GNN-based methods (e.g., MVIN, CKAN) have better performance than path-based (MCRec) and embedding-based (CKE) methods. Compared with the latter two which use semantic representation and semantic connection information independently, GNN-based methods capture and integrate the two aspects between entities. Since the key to recommendation accuracy is to capture the similarity between users and items, the GNN-based models integrate semantic knowledge and connection information through multi-hop neighborhood aggregation and are reflected in the learning vector space. When multi-aspect information is integrated into the vector space, the model captures the relations between users and items more effectively and obtains better recommendation results.
- In most cases, contrastive learning methods have achieved the relatively best performance. Since as an auxiliary learning task, contrastive learning obtains self-supervised signals to guide the main learning tasks by enhancing the data and comparing different structures. Different from the GNN-based methods, contrastive learning methods build sufficient negative samples among the augmented views, thereby enhancing the representation learning ability of the model for better performance. Our model achieves significant performance improvement based on LightGCN, which proves the sparsity of previous supervised signals on user-item interaction. As an additional supervision signal supplement, our method alleviates the long-tail distribution problem and empowers representation learning of traditional recommendation models.

5.3 Ablation study (RQ2)

In this section, a series of ablation experiments are implemented to investigate the benefits and effectiveness of the proposed modules. We generate two variant models by simplifying the key components in BUCL, which are BUCL w/o GE (Graph Embedding) and BUCL w/o UO (User-Oriented);

5.3.1 BUCL w/o GE

As previously mentioned, two graph embedding methods TransR and TATEC are applied to excavate the factual and semantic information, providing refined features for subsequent operations. In order to verify whether the module is effective, we directly initialize node representation with Xavier initialization method. From the Results in Table 3, after removing the graph embedding method, NDCG@20 decreases by 2.75%, 1.45%, and 6.89% respectively. As well, NDCG@20 drops more in the MIND dataset than BUCL w/o UO, which we suspect is the reason the MIND dataset is related to the news article. The knowledge in news articles has richer factual and semantic information than general KG, and the unprocessed input of the Xavier initialization method increases the difficulty of training the model and hurts performance.

5.3.2 BUCL w/o UO

We propose user-oriented item quality estimation method that considers the user's perspective bias to determine the direction of enhancement. To verify the effectiveness of this module, we directly perform min-max normalization for quality score **c** to obtain the sampling probabilities. When ordinary normalized probability is used instead of user-oriented method, NDCG@20 decreases by 2.55%, 2.43%, and 4.70%, respectively. At the same time, we observed that the simplification of the UO module will bring greater losses in most cases, which verified that the differentiated views of users fully consider the preferences of different users and improve the rationality and robustness of the comparison subgraphs.

5.4 Benefits of BUCL (RQ3)

5.4.1 Impact of noise

Since the KG as side information also brings noise, we judge the robustness of BUCL by randomly adding some triples in KG according to different proportions. Figure 3 shows the results of different models with different noise ratios. Obviously, as the proportion of introduced noise increases, BUCL maintains a slower rate of decline and continues to outperform other contrastive learning and KG-based models, which proves that our model is robust enough to resist noise perturbation. Part of the reason for this result is that we introduced the concept of user perspective bias. Other models basically do not filter noise in the KG or filter too coarsely. BUCL considers the different definitions of noise by users' preferences, and models users' preferences to construct a differentiated augmented subgraph. This process effectively suppresses noise effects and provides robust self-supervised signals.

Table 3 Ablation experiment results

dataset / model	Yelp2018		Amazon-Book		MIND	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BUCL	**0.0801**	**0.0523**	**0.1561**	**0.0841**	**0.0973**	**0.0512**
BUCL w/o GE	0.0785	0.0509	0.1546	0.0829	0.0942	0.0479
BUCL w/o UO	0.0776	0.0510	0.1537	0.0821	0.0954	0.0489

Bold represents the best performance of all methods

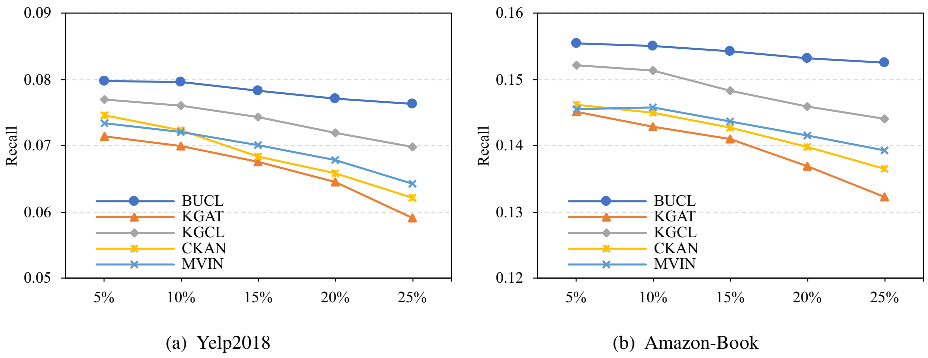

Fig. 3 Impact of noise analysis

5.4.2 Long-tail recommendation

As mentioned above, item nodes in the KG show an obvious long-tail distribution, we demand to verify whether BUCL alleviates this problem. All items are divided into five groups according to their frequency. The larger the group ID is, the higher the item frequency is. The results of the comparison with the other baseline models are shown in Fig. 4. We observed that in the long-tail distribution scenario, our model achieves the best results in most of the groups. In particular, in the case of low item exposure, our model significantly outperforms other models, indicating that BUCL obtains robust representations even with fewer connected edges. Specifically, BUCL fully exploits the factual and semantic information contained in KG and constructs refined feature vectors. The vector space where these feature vectors are located enables the imbalanced items with insufficient representation to learn high-quality features and provide more reasonable self-supervised signals.

5.5 Parameter setting effect (RQ4)

In this section, we study the impact of different hyperparameter settings on model performance, including temperature parameter τ and contrastive loss regularization coefficient λ_1.

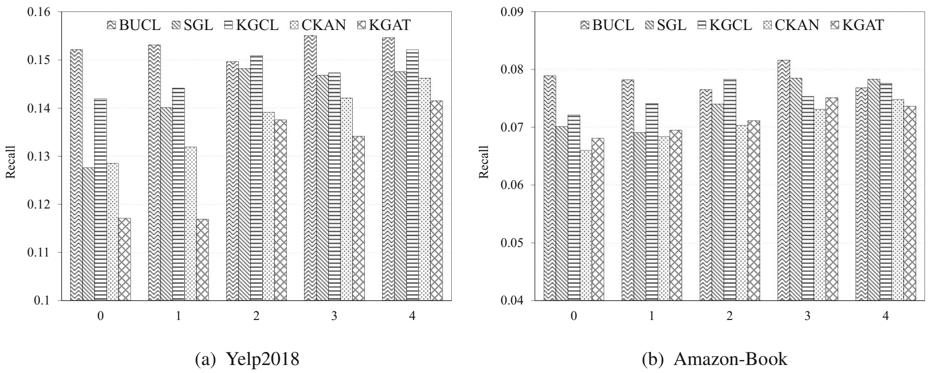

Fig. 4 Long-tail distribution analysis

5.5.1 Effect of temperature τ

In the InfoNCE loss, the temperature parameter controls the discrimination of the model to the negative samples. If the setting is too large, the model treats all the negative samples equally, but if the setting is too small, the model pays too much attention to the negative samples which are difficult to distinguish. Therefore, an appropriate temperature parameter is directly related to the final performance of the model. As shown in Fig. 5, we observed that too large or too small temperature parameters would limit the performance of the model. After our experiment, $\tau = 0.2$ in Amazon-book and $\tau = 0.3$ in MIND are the appropriate choices.

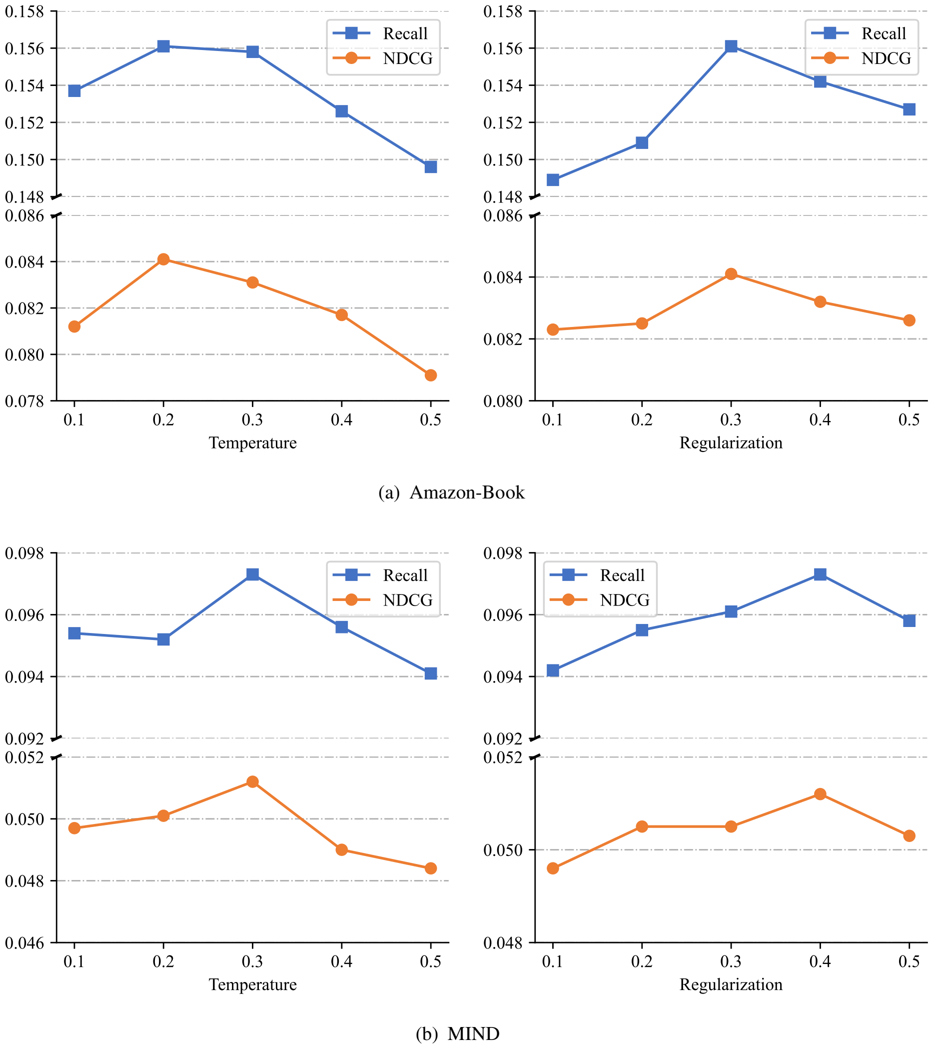

Fig. 5 Parameter sensitive analysis on datasets Amazon-Book and MIND

5.5.2 Effect of regularization coefficient λ_1

Since the contrastive loss and the BPR loss are jointly optimized, the hyperparameter λ_1 is used to balance the composition of the objective function. As shown in Fig. 5, we observed that $\lambda_1 = 2e^{-3}$ in Amazon-Book and $\lambda_1 = 2e^{-4}$ in MIND are suitable balanced points and obtain the best performance.

6 Conclusion

In this paper, we propose BUCL, a model based on the contrastive learning framework, which can alleviate the noise interaction and long-tail distribution issues. BUCL framework applies graph embedding methods with different focus points to fully mine the factual and semantic information contained in KG that is beneficial to representation learning. At the same time, we design an item quality estimation method based on users' perspectives. Consider users' perspectives bias for noise reduction to build robust differentiated subgraphs of users. As a self-supervised learning framework, BUCL can be jointly optimized with the KG-based models as an auxiliary learning task to enhance representation ability. We have conducted a lot of experiments to prove the effectiveness and robustness of our model, which can significantly improve recommendation performance.

The limitation of our method is that it only models singular type interactions (such as purchase, click) from the user's point of view, lacking the refined modeling of user multi-behavior preference. However, since our proposed components are generalized, these methods are also applicable to similar scenarios. For future work, we plan to introduce KG into multi-behavior recommendation, and mine the semantics of KG to construct user-oriented multi-behavior views to capture users' fine-grained preferences for the target behavior.

Acknowledgements We thank the instructor Li Bohan for our guidance and the funds for our support.

Author Contributions Yi Liu gave his contribution to the construction of the overall model, the design and exploration of related experiments, and completed the writing of the manuscript. Hongrui Xuan participated in model design, exploration of experimental results, and review of the first manuscript. Bohan Li provided the idea through constructive discussions and gave the formulation of the overall research goals.

Funding This work was supported in part by the National Natural Science Foundation of China under Grant 62172351, the "14th Five-Year Plan" Civil Aerospace Pre-Research Project of China under Grant D020101, the Fund of Prospective Layout of Scientific Research for NUAA(Nanjing University of Aeronautics and Astronautics, and the Practice Innovation Program of NUAA (Grant No. xcxjh20221605).

Data Availability The data is available at https://github.com/RUCAIBox/RecSysDatasets.

Declarations

Competing interests The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

Atas, M., Felfernig, A., Polat-Erdeniz, S., & et al. (2021). Towards psychology-aware preference construction in recommender systems: Overview and research issues. *Journal of Intelligent Information Systems*, 1–23. https://doi.org/10.1007/s10844-021-00674-5.

Cao, Y., Wang, X., He, X., & et al (2019). Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The World Wide Web conference* (pp. 151–161). https://doi.org/10.1145/3308558.3313705.

Chen, T., Kornblith, S., Norouzi, M., & et al (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning, PMLR* (pp. 1597–1607). https://doi.org/10.48550/arXiv.2002.05709.

Chen, Y. (2015). Convolutional neural network for sentence classification. Master's thesis, University of Waterloo.

Cole, E., Yang, X., Wilber, K., & et al (2022). When does contrastive visual representation learning work? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14,755–14,764). https://doi.org/10.48550/arXiv.2105.05837.

Dai, S., Yu, Y., Fan, H., & Dong, J (2022). Spatio-temporal representation learning with social tie for personalized POI recommendation. *Data Science and Engineering*, *7*(1), 44–56. Springer, https://doi.org/10.1007/s41019-022-00180-w.

Desrosiers, C., & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. *Recommender Systems Handbook*, 107–144. https://doi.org/10.1007/978-0-387-85820-3_4.

Gao, C., Wang, X., He, X., & et al (2022). Graph neural networks for recommender system. In *Proceedings of the 15th ACM international conference on Web search and data mining* (pp. 1623–1625). https://doi.org/10.1145/3488560.3501396.

García-Durán, A., Bordes, A., & Usunier, N. (2014). Effective blending of two and three-way interactions for modeling multi-relational data. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 434–449). Springer. https://doi.org/10.1007/978-3-662-44848-9_28.

Ge, S., Wu, C., Wu, F., & et al. (2020). Graph enhanced representation learning for news recommendation. In *Proceedings of the Web conference*, (Vol. 2020 pp. 2863–2869). https://doi.org/10.1145/3366423.3380050.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings* (pp. 249–256).

Guo, W., Zhang, C., He, Z., & et al. (2022). MISS: Multi-interest self-supervised learning framework for click-through rate prediction. In *2022 IEEE 38th international conference on data engineering (ICDE)* (pp. 727–740). IEEE. https://doi.org/10.1109/ICDE53745.2022.00059.

He, K., Fan, H., Wu, Y., & et al. (2020a). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9729–9738).

He, X., Deng, K., Wang, X., & et al. (2020b). Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in Information Retrieval* (pp. 639–648). https://doi.org/10.1145/3397271.3401063.

Hu, B., Shi, C., Zhao, W. X., & et al. (2018). Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1531–1540). https://doi.org/10.1145/3219819.3219965.

Jiao, Y., Xiong, Y., Zhang, J., & et al. (2020). Sub-graph contrast for scalable self-supervised graph representation learning. In *2020 IEEE international conference on data mining (ICDM)* (pp. 222–231). IEEE. https://doi.org/10.1109/ICDM50108.2020.00031.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:14126980, https://doi.org/10.48550/arXiv.1412.6980.

Koren, Y., Rendle, S., & Bell, R. (2022). Advances in collaborative filtering. *Recommender Systems Handbook*, pp. 91–142. https://doi.org/10.1007/978-1-0716-2197-4_3.

Lei, J., Li, Y., Yang, S., & et al. (2022). Two-stage sequential recommendation for side information fusion and long-term and short-term preferences modeling. *Journal of Intelligent Information Systems*, 657–677. https://doi.org/10.1007/s10844-022-00723-7.

Lin, Y., Liu, Z., Sun, M., & et al. (2015). Learning entity and relation embeddings for knowledge graph completion. In *29th AAAI conference on artificial intelligence*.

Liu, Z., Ma, Y., Ouyang, Y., & et al. (2021). Contrastive learning for recommender system. arXiv:210101317, https://doi.org/10.48550/arXiv.2101.01317.

Panda, D. K., & Ray, S. (2022). Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. *Journal of Intelligent Information Systems*, 1–26. https://doi.org/10.1007/s10844-022-00698-5.

Qiu, R., Huang, Z., Yin, H., & et al. (2022). Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the 15th ACM international conference on Web search and data mining* (pp. 813–823). https://doi.org/10.1145/3488560.3498433.

Rendle, S., Freudenthaler, C., Gantner, Z., & et al. (2014). Bayesian personalized ranking from implicit feedback. In *Proceedings of uncertainty in artificial intelligence* (pp. 452–461).

Shi, C., Hu, B., Zhao, W. X., & et al. (2018). Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *31*(2), 357–370. IEEE, https://doi.org/10.1109/TKDE.2018.2833443.

Sun, Z., Yang, J., Zhang, J., & et al. (2018). Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM conference on recommender systems* (pp. 297–305). https://doi.org/10.1145/3240323.3240361.

Wang, C., Liang, Y., Liu, Z., & et al. (2021). Pre-training graph neural network for cross domain recommendation. In *2021 IEEE 3rd international conference on cognitive machine intelligence (CogMI)* (pp. 140–145). IEEE.

Wang, H., Zhang, F., Wang, J., & et al (2018a). RippleNet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management* (pp. 417–426). https://doi.org/10.1145/3269206.3271739.

Wang, H., Zhang, F., Xie, X., & et al. (2018b). DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web conference* (pp. 1835–1844). https://doi.org/10.1145/3178876.3186175.

Wang, H., Zhao, M., Xie, X., & et al. (2019a). Knowledge graph convolutional networks for recommender systems. In *The World Wide Web conference* (pp. 3307–3313). https://doi.org/10.1145/3308558.3313417.

Wang, J., Huang, P., Zhao, H., & et al. (2018c). Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 839–848). https://doi.org/10.48550/arXiv.1803.02349.

Wang, Q., Mao, Z., Wang, B., & Guo, L (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724–2743. IEEE, https://doi.org/10.1109/TKDE.2017.2754499.

Wang, X., He, X., Cao, Y., & et al. (2019b). KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 950–958). https://doi.org/10.1145/3292500.3330989.

Wang, X., He, X., Wang, M., & et al. (2019c). Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 165–174). https://doi.org/10.1145/3331184.3331267.

Wang, X., Wang, D., Xu, C., & et al. (2019d). Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 5329–5336).

Wang, Z., Lin, G., Tan, H., & et al. (2020). CKAN: collaborative knowledge-aware attentive network for recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval* (pp. 219–228). https://doi.org/10.1145/3397271.3401141.

Wu, J., Wang, X., Feng, F., & et al. (2021). Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 726–735). https://doi.org/10.1145/3404835.3462862.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S.Yu (2020a). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24. IEEE, https://doi.org/10.1109/TNNLS.2020.2978386.

Wu, Z., Wang, S., Gu, J., & et al. (2020b). Clear: Contrastive learning for sentence representation. arXiv:201215466, https://doi.org/10.48550/arXiv.2012.15466.

Xia, L., Huang, C., Xu, Y., & et al. (2022). Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 70–79). https://doi.org/10.1145/3477495.3532058.

Xian, Y., Fu, Z., Muthukrishnan, S., & et al. (2019). Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 285–294).

Xie, R., Liu, Q., Wang, L., & et al. (2021). Contrastive cross-domain recommendation in matching. arXiv:211200999, https://doi.org/10.1145/3534678.3539125.

Yan, H., Yang, J., & Wan, J. (2020). KnowIME: a system to construct a knowledge graph for intelligent manufacturing equipment. *IEEE Access*, *8*, 41,805–41,813. IEEE, https://doi.org/10.1109/ACCESS.2020.2977136.

Yang, Y., Huang, C., Xia, L., & et al. (2022). Knowledge graph contrastive learning for recommendation. arXiv:220500976, https://doi.org/10.48550/arXiv.2205.00976.

Yao, T., Yi, X., Cheng, D. Z., & et al. (2021). Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 4321–4330). https://doi.org/10.1145/3459637.3481952.

Yu, J., Yin, H., Xia, X., & et al. (2021). Graph augmentation-free contrastive learning for recommendation. arXiv:211208679, https://doi.org/10.48550/arXiv.2112.08679.

Yu, J., Yin, H., Xia, X., & et al. (2022). Self-supervised learning for recommender systems: A survey. arXiv:220315876, https://doi.org/10.48550/arXiv.2203.15876.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. arXiv:14092329, https://doi.org/10.48550/arXiv.1409.2329.

Zhang, F., Yuan, N. J., Lian, D., & et al. (2016). Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 353–362). https://doi.org/10.1145/2939672.2939673.

Zhang, J., Gao, M., Yu, J., & et al. (2021). Double-scale self-supervised hypergraph learning for group recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 2557–2567). https://doi.org/10.1145/3459637.3482426.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81. Elsevier, https://doi.org/10.1016/j.aiopen.2021.01.001.

Zou, D., Wei, W., Mao, X. L., & et al. (2022). Multi-level cross-view contrastive learning for knowledge-aware recommender system. arXiv:220408807, https://doi.org/10.48550/arXiv.2204.08807.