Knowledge Graph Preference Contrastive Learning for Recommendation

Junze Zhu
Wuhan University
zjzzjz@whu.edu.cn

Zhongyi Hu
Wuhan University
zhongyi.hu@whu.edu.cn

Fan Zhang
Wuhan University
frankyzf94@gmail.com

Abstract

Recent research has incorporated knowledge graphs to mitigate the issue of data sparsity in recommendations. However, while leveraging the rich information from knowledge graphs exhibits promising performance enhancements, it also introduces noise that potentially disrupts collaborative signals. To overcome this problem, we propose the Knowledge Graph Preference Contrastive Learning for Recommendation, namely KPCL. The preference learning method and rationale attention mechanism are designed to explicitly track collaborative signals and identify informative knowledge connections from both macro and micro perspectives. Specifically, preference learning is used to alleviate semantic dissonance in knowledge embeddings by exploring intent correlations in user-item interaction history, while the rationale attention restructures the knowledge graph by eliminating knowledge triplets with low attention scores as noise. By aggregating the information in the knowledge graph through the selected knowledge triplets, the task-unrelated noise presented would be filtered out, leading to enhanced performance for the knowledge-aware recommender system. Experimental results on three benchmark datasets demonstrate the superiority of KPCL over the state-of-the-art methods. The implementations for KPCL are available at https://github.com/HuiCir/KPCL.

1 Introduction

In the era of information overload, recommender systems have become critical personalized information retrieval tools [1–3]. Collaborative Filtering (CF) has been widely adopted as an effective recommendation approach [4–8]. However, CF systems face challenges due to their heavy dependence on historical user interaction data, resulting in issues of sparsity and cold-start [9–11]. To overcome these data-related problems, Knowledge Graphs (KG) have been incorporated into the recommendation tasks as side information to enhance the user-item representation by encoding additional entity and relation information from KG [12, 13]. Previous studies have explored the integration of KG into CF systems to obtain richer side information and potential multi-hop high-order proximity [14–17]. Not only improving the recommendation performance, KG has also demonstrated great potential in enhancing the explainability of recommendation [18, 19].

The main task of knowledge-aware recommendation is to derive high-quality user-item representations from structural knowledge, with the aim of improving recommendation performance. Early research [14, 18, 20] has mainly focused on the embeddings of KG and treated them as side information to enhance item representations. To capture high-order connectivity in KG, some path-based approaches [12] have attempted to extract semantically meaningful meta-paths from the KG and learn more complex user-item representations along these meta-paths [19, 21, 22]. However, most path-based models are restricted by various issues like labor-intensive feature engineering [19] and unstable performance [23]. To combine the strengths of embedding-based and path-based methods seamlessly and automatically, Graph Neural Networks (GNN) [24–27] have been introduced into recent studies [28, 29] on knowledge-aware recommendation. By constructing a user-item interaction graph and cooperating with structured knowledge from KG, propagation-based methods effectively derive multi-

Junze. Zhu et al., Knowledge Graph Preference Contrastive Learning for Recommendation. *Proceedings of the Third Learning on Graphs Conference (LoG 2024)*, PMLR 269, Virtual Event, November 26–29, 2024.

hop high-order representations through recursive propagation and aggregation of GNNs. Benefiting from the integration of connectivity, these propagation-based models have achieved state-of-the-art performance for recommendation, such as KGAT [15], KGCN [30], KTUP [12] and KGIN [31].

Despite the potential benefits of utilizing KGs to enhance item representations, real-world KGs are often noisy and contain task-irrelevant entities. This can lead to suboptimal performance in recommendation tasks [32–35], particularly when the scale of the KG is large and the noise signals outnumber the effective signals required for CF tasks [36]. To address this issue, recent studies have proposed various self-supervised techniques to alleviate noise in KGs. For instance, KGCL [37] introduces contrastive learning based on stochastic graph augmentation to address noise issues in KGs. To avoid the uncertainty of simple random augmentation, KACL [38] proposes two learnable augmentation generators to perform cross-view contrastive learning adaptively. Yang et al. [39] propose a self-supervised method to perform rationalization of KG to capture the collaborative signal implicitly. KGIL [40] introduces the principle of invariance to the knowledge-aware recommendation, which aims to discern and harness the task-relevant knowledge connections. On the other hand, some studies employ generative techniques to overcome the noise problem. DiffKG [41] integrates a diffusion model with a data augmentation paradigm, enabling robust knowledge graph representation learning to resist noise. Tang et al. [42] design a knowledge generator to generate attributes for items by exploring their mutual information correlations and semantic correlations. However, we argue that these methods do not explicitly consider the semantic dissonance between additional knowledge and user preferences in collaborative signals. The knowledge that meets user preferences is more suitable for recommendation tasks. Therefore, the present study proposes a preference contrastive learning method to explicitly track collaborative signals from interaction history.

The knowledge graph introduces side information using the connections between entities. If we aggregate all these connections indifferently, it could interfere with the learned item representations due to irrelevant noisy entities. Given a target user, the collaborative signals are extracted from the interaction history with peers sharing the same interests. By mining user preferences from CF signals, the irrelevant entities serving as negative samples will be distributed less attention than other candidates. Thereby, the knowledge aggregation in KG can be conducted optionally by highlighting preference-related knowledge triplets.

Overall, it is important to discover the preference by tracking the CF signals in the knowledge graph recommender systems. To tackle this challenge, we propose a **K**nowledge Graph **P**reference **C**ontrastive **L**earning for Recommendation, namely KPCL. KPCL constructs a preference learning mechanism that matches the preference of collaborative signals with the relation entities in KG. It allows the collaborative signals, expressed as a reward to different relations, to guide the hierarchical aggregation process explicitly. Consequently, KPCL can strategize a heterogeneous aggregator that is sensitive to collaborative signals, thereby improving the performance of recommendation tasks. Finally, we filter out edges with less contribution and perform preference-driven aggregation of information to generate the final embedding representation.

To summarize, the main contributions of this paper are three folds:

(1) We design a relation rank preference learning to effectively catch user's intentions by explicitly aligning them with collaborative signals. This mechanism optimizes the relation weights globally based on different preference requirements, which improves the embedding representations of entities and adapts them to downstream recommendation tasks appropriately.

(2) The rationale attention mechanism allows us to individually extract relevant triplets by distinguishing between noise and critical task-related connections that significantly contribute to recommendation tasks. This method allows the model to refine the KG by diminishing the impact of noise, resulting in a more streamlined and accurate knowledge base for recommendation tasks.

(3) To validate the effectiveness of our proposed KPCL method, we conduct extensive experiments using three real-world datasets. The results demonstrate that the proposed model outperforms the state-of-the-art recommendation methods. Particularly in high-density knowledge graphs, KPCL exhibits superior capabilities in more precise rationalization and target noise connections.

2 PROBLEM FORMULATION

2.1 User-Item Interaction Graph

In recommendation, the signal of a user's preference is a form of implicit feedback extracted from historical user-item interactions [43]. Let $\mathcal{U}$ be the set of users and $\mathcal{I}$ the set of items. $u \in \mathcal{U}$ and $i \in \mathcal{I}$ denote a single user and item, respectively. The interaction between u and i is denoted by y_{ui}, with $y_{ui} = 1$ if user u interact with the item i, and $y_{ui} = 0$, otherwise. Thereby, the user-item interaction graph can be expressed as $\mathcal{V} = \{(u, y_{ui}, i)|u \in \mathcal{U}, i \in \mathcal{I}\}$, where (u, y_{ui}, i) is a triplet denoting a specific user-item interaction.

2.2 Knowledge Graph

KG stores the entity-relation-entity structured information of real-world facts, in the form of a heterogeneous graph or heterogeneous information network [44]. Let $\mathcal{E}$ be a set of entities and $\mathcal{R}$ be the relation set. The KG, denoted by $\mathcal{G} = \{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{R}\}$, is a collection of triplets, with a triplet (h, r, t) representing the relation between the head entity h and tail entity t. To improve the quality of recommendation, Knowledge-aware recommendations utilize KG as side information, with $\mathcal{I} \subset \mathcal{E}$.

2.3 Task Description

The task of this knowledge-aware recommendation can be formally defined as follows: Provided with the user-item interaction data $\mathcal{V}$ and the KG $\mathcal{G}$, the goal is to learn a function $\mathcal{F}(u, i|\mathcal{V}, \mathcal{G}, \Theta)$ that can predict the preference of user u interacting with item i, where Θ are learnable parameters.

3 Methodology

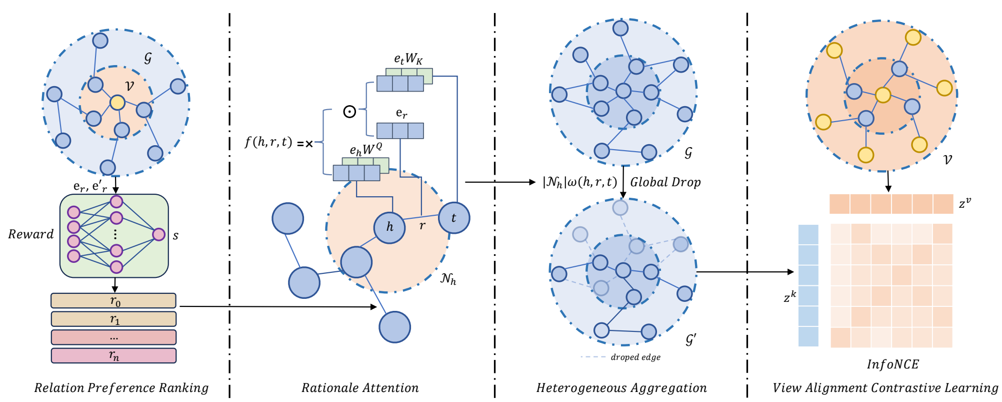

Figure 1: The overall framework of KPCL

The proposed KPCL is depicted in Figure 1, and the details are presented in this section.

3.1 Relation Preference Learning

By adjusting individual weights based on attention mechanism from a micro perspective, previous studies [15, 39, 40] optimized the KG representation with less noise. On the contrary, this study takes a more macroscopic perspective, aligning collaborative signals by optimizing the embeddings of relations, while optimizing triplet individuals globally. When we consider the head-tail connection categories, the role it plays in a knowledge triplet, this will manifest as the embeddings of relation. Inspired by the recent success of RLHF (Reinforcement Learning from Human Feedback) [45, 46], the interaction of humans can be used to capture user's preferences and further guide model's optimization.

In this study, we introduce Bradley-Terry model as reward model for relation preference learning from user intention information in the user-item interaction graph. The preference rewards are used to catch the collaborative filtering signals and guide KG optimization. Formally, we define preference rewards embeddings $\mathcal{P}$ in the same space as R. To establish the bridge between preferences and relations, the formula for calculating preference relation ranking scores is as follows:

$$s = \sum_{p\in\mathcal{P}, r\in\mathcal{R}} exp(\mathbf{e}_p \cdot \mathbf{e}_r); s' = \sum_{p\in\mathcal{P}, r'\in\mathcal{R}'} exp(\mathbf{e}_p \cdot \mathbf{e}_{r'}), \tag{1}$$

where, $\mathbf{e}_p$ and $\mathbf{e}_r$ represent the embedding of preference reward and relation, respectively. $r \in \mathcal{R}$ represents the original relation used to calculate the ranking score for matching preferences, denoted as s. While $\mathbf{e}_{r'}$ represents the masked relation embeddings used to calculate the score for non-matching preferences, denoted as s'. For $\mathcal{R}'$, non-existent relations in the input user interaction information would be masked. Thereby, we can define the loss function of preferences by the following formula:

$$\mathcal{L}_p = -log\frac{s}{\alpha s + s'}, \tag{2}$$

By extending the Bradley-Terry model, we introduce an additional hyperparameter α to adjust the focus of preferences to positive and negative samples. By doing so, the preference information $\mathcal{P}$ from user intents in collaborative signals can be delivered to knowledge triplets. The preference-driven relation embeddings will guide the further optimization of the triplet in KG.

3.2 Rationale Attention

To extract essential semantic information for collaborative interactions from the complex knowledge graph, we introduce an attentive weighting function that learns the probability of how a knowledge triplet can reflect the underlying rationale for collaborative signals. Following previous works [15, 39, 47], which discerns the importance of heterogeneous relations in KG, the attention calculation function $f(h, r, t)$ is defined by the following formula:

$$f(h, r, t) = \frac{\mathbf{e}_h\mathbf{W}^Q \cdot (\mathbf{e}_t\mathbf{W}^K \odot \mathbf{e}_r)^\top}{\sqrt{d}}, \tag{3}$$

where $\mathbf{e}_h, \mathbf{e}_r$ and $\mathbf{e}_t$ represent the embeddings for the head, relation, and tail of the knowledge triplet. $\mathbf{W}^k, \mathbf{W}^Q \in \mathbb{R}^{d\times d}$ are learnable weights for graph attention, where d is the hidden dimensionality. The Hadamard product of the relation $\mathbf{e}^r$ and the tail entity $\mathbf{e}_t$ denotes the rotation of the tail entity embedding $\mathbf{e}_t$ to the latent space of relation $\mathbf{e}_r$ [39, 48].

Furthermore, a multi-view criterion, formulated by Eq.(4), was designed to assess the rationale importance of edges from both global and local viewpoints [39].

$$\omega(h, r, t) = |\mathcal{N}_h| \cdot \frac{exp(f(h, r, t))}{\sum_{(h,r',t')\in\mathcal{N}_h} exp(f(h, r', t'))}, \tag{4}$$

where $\mathcal{N}_h$ is the neighbors of the head entity h. $|\mathcal{N}_h|$ is the number of neighbors of h. The right part of Eq.(4) is the rationale score, normalized by the softmax function, for estimating the local importance of the triplet among all edges connected to the same head entity h. By multiplying the local importance with $|\mathcal{N}_h|$, the rational score is globally weighted based on the number of degrees of head entity. Next, we construct a low-contribution noise set of edges based on $\omega(h, r, t)$ with a ratio $\gamma \in (0, 1)$. By removing these edges from the original knowledge graph $\mathcal{G}$, we create a rationale-aware augmented knowledge graph, denoted as $\mathcal{G}'$.

3.3 Heterogeneous Aggregation

With the above processing, KPCL could pay more attention to the knowledge connections with higher rationale scores, which enables the model to infer the intents of collaborative signals. Thus, by weighting knowledge triplets with their corresponding knowledge rationale scores, the knowledge aggregator can be defined as:

$$\mathbf{e}_h^{(l)} = \frac{1}{|\mathcal{N}_h'|}\sum_{(h,r,t)\in\mathcal{N}_h'} \omega(h, r, t)\mathbf{e}_r \odot \mathbf{e}_t^{(l-1)}, \tag{5}$$

where l denotes the certain layer of the aggregator. $\mathcal{N}_h'$ is an ego graph of first-order neighbors sampled from the augmented KG $\mathcal{G}'$. The attention scores of the preserved triplets do not need to be recalculated, as the discarded structures have such low scores that their minor variations can be ignored.

As for the entity representations in the user-item graph, considering items are a subset of knowledge entities, the item embeddings $\mathbf{e}_i$ can be collected from the trained entities embeddings $\mathbf{e}_h$. Regarding the users' embeddings, a neighbor aggregation method defined by the following formulas is used.

$$\mathbf{e}_u^{(l)} = \frac{1}{|\mathcal{N}_u|} \sum_{i \in \mathcal{N}_u} \mathbf{e}_i^{(l-1)}; \mathbf{e}_u = \sum_{l}^{L} \mathbf{e}_u^{(l)} \tag{6}$$

where $\mathbf{e}_u$ and $\mathbf{e}_i$ are the embeddings of user and item, respectively. L is the number of layers in the GNN aggregation. $\mathcal{N}_u \subset \mathcal{V}$ is a set of neighboring nodes of the items i. These neighboring nodes have connections through their interaction histories with the user u. Similarly, the item embedding $\mathbf{e}_i$ in the user-item graph can also be calculated by a method like Eq.(6). This allows us to model collaborative signals between users and items by aggregating the embeddings of the neighboring items through the user-item interaction histories.

3.4 View Alignment Contrastive Learning

To capture the view-specific node representations, LightGCN [49] is used to iteratively model the interactions and capture high-order information on the user-item graph $\mathcal{V}$. Formally, LightGCN could be implemented as:

$$\mathbf{x}_u^{(l)} = \sum_{i \in \mathcal{N}_u} \frac{\mathbf{x}_i^{(l-1)}}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}; \mathbf{x}_i^{(l)} = \sum_{u \in \mathcal{N}_i} \frac{\mathbf{x}_u^{(l-1)}}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}. \tag{7}$$

Here, $\mathbf{x}_u^{(l)}$ and $\mathbf{x}_i^{(l)}$ are the embeddings of user u and item i in the l-th layer, respectively. $\mathcal{N}_u$ and $\mathcal{N}_i$ represent the connected items and users of u and i, respectively. Consequently, the representations of items in the collaborative filtering view, denoted as $\mathbf{x}^v$, are obtained by summing up the embeddings of L layers:

$$\mathbf{x}^v = \sum_{l}^{L} \mathbf{x}_u^{(l)}; \mathbf{x}^k = \sum_{l}^{L} \mathbf{e}_i^{(l)}, \tag{8}$$

Similarly, the embeddings of entities, denoted as $\mathbf{x}^k$, in KG view are calculated by the aggregation on the augmented graph $\mathcal{G}'$.

As stated in previous studies [37, 39, 50], cross-view contrastive learning is crucial for knowledge-aware recommender. Therefore, we adopt the contrastive learning method to align the representations of the knowledge graph and collaborative signals. Specifically, we first map the embeddings $\mathbf{x}^v$ and $\mathbf{x}^k$ from different spaces into the same latent space as $\mathbf{z}^v$ and $\mathbf{z}^k$ by using a two-layer MLP. The loss function $\mathcal{L}_c$ is then calculated using the InfoNCE paradigm [51, 52] to align the collaborative relational signals and knowledge graph signals.

$$\mathcal{L}_c = \sum_{i \in \mathcal{I}} -log \frac{exp(s(\mathbf{z}_i^v, \mathbf{z}_i^k)/\tau)}{\sum_{j \in \mathcal{J}} (exp(s(\mathbf{z}_i^v, \mathbf{z}_i^k)/\tau) + exp(s(\mathbf{z}_j^v, \mathbf{z}_j^k)/\tau))}. \tag{9}$$

Here, τ is a hyperparameter called temperature that controls the focus on hard negatives. Besides, a candidate set of negative sampling $\mathcal{J} = \{i, i'\}$ is constructed, including positive and other non-positive items. The randomly selected one from this candidate set is applied as the negative sampling result j to participate in contrastive learning. The $s(\cdot)$ is used to estimate the matching degree of two representations, and is set as a cosine similarity function in this study.

3.5 Model Optimization

Given the final representations of a user u and an item i, the inner product $y(u, i) = \mathbf{e}_u^\top \mathbf{e}_i$ can be employed to measure the user's preference on the item. Then, the widely used BPR loss [43] is used to optimize the parameters in the model:

$$\mathcal{L}_{rec} = \sum_{(u,i,i') \in \mathcal{S}} -log\sigma(y(u, i) - y(u, i')), \tag{10}$$

where $\mathcal{S}$ represents the training set for training the model. $i \in \mathcal{N}_u$ is the ground-truth for user u. $i' \notin \mathcal{N}_u$ is a stochastic negative interaction.

Finally, a joint learning for KPCL is designed, which consists of three tasks, i.e., the main recommendation task, preference learning task, and contrastive learning task. Together with L2 regularization, the loss function of the joint learning is defined as follows.

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_1 \mathcal{L}_p + \lambda_2 \mathcal{L}_c + ||\Theta||_2^2. \tag{11}$$

Here, λ_1 and λ_2 represent the weight of the relation preference learning and view alignment contrastive learning tasks, respectively. Θ represents the learnable model parameters.

3.6 Model Complexity

We analyze the time complexity of our KPCL framework from three key components, i.e., aggregation scheme, preference learning, and contrastive learning. In the aggregation over User-Item Interaction Graph, the complexity of user representations is $O(L \times |\mathcal{V}| \times d)$, where L, $|\mathcal{V}|$, and d denote the number of layers, the number of triplets in $\mathcal{V}$, and embedding size, respectively. In the aggregation over KG, the time cost of updating entity representations is $O(L \times |\mathcal{G}| \times d)$, where $|\mathcal{G}|$ is the number of KG triplets. Then, The relation preference learning module takes $O(|\mathcal{V}| \times d^2)$ time. As for the contrastive learning, the time complexity to calculate InfoNCE loss is $O((|\mathcal{U}| + |\mathcal{I}|) \times d)$, where $|\mathcal{U}|$ and $|\mathcal{I}|$ denote the number of users and items in $\mathcal{U}$ and $\mathcal{I}$. Under the same experimental settings, KPCL has comparable complexity to other baselines[31, 39].

4 EXPERIMENTS

4.1 Dataset Description

To evaluate the effectiveness of KPCL, we perform experiments using three benchmark datasets from small to large scale:

- **Movielens** [38] is a classic movie recommendation dataset gathered from a research-focused website of the same name.
- **Alibaba-iFashion** [53] is a collection of fashion outfits from Alibaba's e-commerce platform, documenting user-outfit click history.
- **Amazon-Book** [54] is a widely used dataset for product recommendation selected from Amazon-review.

Both of the datasets are publicly available and have been used in previous studies on knowledge-based recommendation. The statistics of three datasets are presented in the appendix.

4.2 Evaluation Metrics

We adopt the commonly used all-ranking strategy metrics in the top-K recommendation and preference ranking tasks: Recall@K and NDCG@K, with K set to 20. Recall@K shows the proportion of their rated items that are included in the top K recommended items. NDCG@K represents the normalized discounted cumulative gain at K, which accounts for the positioning of correctly recommended items.

4.3 Overall Performance Comparison

We conduct benchmark evaluations comparing KPCL and other baseline models from a variety of research perspectives.

General Collaborative Filtering

- **BPR** [43] is a matrix factorization model that utilizes pairwise ranking loss and is based on implicit feedback, focusing solely on user-item interactions.
- **NFM** [55] incorporates MLP into matrix factorization to learn the non-linear user-item feature interactions.
- **LightGCN** [49] simplifies the convolution operations during the message passing among users and items by removing activation and feature transformation.

Table 1: Evaluation results on Movielens, Alibaba-iFashion and Amazon-book. The best and second-best results are denoted in boldface and borderline.

Model	Movielens		Alibaba-iFashion		Amazon-book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR	0.4114	0.2673	0.0672	0.0406	0.1074	0.0593
NFM	0.3554	0.2146	0.0506	0.0276	0.1033	0.0532
LightGCN	0.4395	0.2932	0.0995	0.0613	0.1396	0.0738
SGL	0.4341	0.2931	0.1113	0.6982	0.1445	0.0766
CKE	0.4094	0.2652	0.0647	0.0390	0.1342	0.0700
KGAT	0.4122	0.2664	0.0711	0.0416	0.1408	0.0744
KGNN-LS	0.4218	0.2741	0.0983	0.0633	0.1362	0.0560
KGIN	0.4601	0.3089	0.1158	0.0721	0.1436	0.0748
KGCL	0.4533	0.3053	0.1146	0.0719	0.1496	<u>0.0793</u>
MCCLK	0.4204	0.2752	0.0985	0.0610	0.1125	0.0633
KGRec	<u>0.4624</u>	<u>0.3097</u>	<u>0.1176</u>	<u>0.0734</u>	<u>0.1513</u>	0.0741
KPCL	**0.4647**	**0.3117**	**0.1181**	**0.0741**	**0.1609**	**0.0859**

- **SGL** [56] introduces a self-supervised learning paradigm to GNN-based recommendation by applying stochastic augmentation on the user-item graph.

Knowledge-aware Recommenders

- **CKE** [14] is an embedding-based approach, which adopts TransR to encode the semantic information of items into the matrix factorization framework.
- **KGAT** [15] constructs a collaborative knowledge graph (CKG) from KG and user-item graph and designs an attentive message passing scheme over CKG.
- **KGNNLS** [28] is a GNN-based model, modelling relations in decay factors, which introduces label smoothing as regularization to force similar user preference weights between neighbors items in the KG.
- **KGIN** [31] is a powerful approach for modelling intents from collaborative interactions, focusing on relations between entities and employing relational path-aware aggregation to encode information from KG.

Self-Supervised Knowledge-aware Recommenders

- **KGCL** [37] incorporates knowledge augmentation into a cross-view contrastive learning approach to reduce the disturbances of noise caused by knowledge overload.
- **MCCLK** [50] is a hierarchical data augmentation contrastive learning framework, which considers both multi-level graph view and structural collaborative semantic views.
- **KGRec** [39] designs a generative task in the form of masking-reconstructing to highlight rationales in the knowledge graph. By rebuilding useful edges serving as rationales, KGRec reconstructs the aggregation path of GNN and learns suitable representations for recommendation systems.

In this Experiment, baseline models are implemented from the original articles. Both the baseline model and ours are trained with the following parameters: batch size 1024, embedding size 64, and learning rate 1e-5. We present the performance comparison of all methods in Table 1, and summarize the results as follows:

The proposed KPCL demonstrates consistent superior performance over all baselines, as evidenced by its higher scores in terms of Recall@20 and NDCG@20 across three datasets. Remarkably, it achieves significant improvements when evaluated on a denser and larger knowledge graph (e.g., Amazon-book), outperforming the strongest baseline model by 6.34% in terms of Recall@20. This

Table 2: Ablation results of KPCL with different variants. The superscript $*$ denotes the largest change in performance. The Relation Preference module and the Rationale Attention module are denoted as RP and RA, respectively.

Ablation	Movielens		Alibaba-iFashion		Amazon-book	
Setting	Recall	NDCG	Recall	NDCG	Recall	NDCG
w/o RP	0.4614*	0.3086*	0.1181	0.0737	0.1434*	0.0748*
w/o RA	0.4627	0.3100	0.1177*	0.0735*	0.1513	0.0797
KPCL	**0.4647**	**0.3117**	**0.1181**	**0.0741**	**0.1609**	**0.0859**

notable performance can be attributed to the ability of KPCL to effectively filter out more noise interfering with collaborative signals than other GNN-based Knowledge-aware baselines. Several key factors contribute to this improvement within KPCL. First, through relation preference learning, KPCL successfully captures the critical knowledge connections that genuinely contribute to the recommendation task. Second, by leveraging rationale attention scores to filter out noise and employing an efficient aggregation path, the performance of GNN is notably enhanced. Lastly, KPCL employs view alignment contrastive learning on the augmented graphs from both KG and CF views, thereby enhancing the uniformity of embedding representation while aligning signals across different views.

It is evident that most knowledge-aware models outperform BPR and NFM. This verifies that incorporating additional information from knowledge graphs effectively alleviates the sparsity issue that commonly exists in collaborative filtering systems. Notably, two notable exceptions, LightGCN and SGL, which are the recent state-of-the-art techniques without considering KG, have outperformed some KG-based models. This demonstrates that simply introducing knowledge graphs does not always guarantee improved performance in recommendation systems. Among the knowledge-aware methods evaluated in our experiment, KGIN stands out as the best model. This model excels in capturing user latent intention by modeling the representation of interactions. Notably, even on small-scale datasets such as MovieLens and Alibaba-iFashion, KGIN achieves better results than some self-supervised models (e.g., MCCLK and KGCL). Turning to our proposed approach, KPCL consistently outperforms all baselines across datasets with varying distribution types, especially for Amazon-Book with larger KG scales. We attribute this improvement primarily to the effective noise-filtering capabilities of KPCL.

Across the three datasets, the improvement of the proposed KPCL is most significant on Amazon-Book. This is in line with our expectations, given that the KG of the Amazon-Book has nearly ten times more knowledge connections compared to the MovieLens and Alibaba-iFashion datasets. While more diverse entities and links in the KG provide more auxiliary information, they also introduce more noise, leading to a substantial impact on recommender systems. Particularly, when dealing with a complex and dense KG with sparse interactions, the noise significantly affects the performance of the recommendation models. Based on the results of t-test on MovieLens and Amazon-Book, the KPCL is statistically different from all the other models with a p-value<0.05. Given that Alibaba-iFashion is the most sparse one, the KPCL is not statistically different and the improvement of KPCL is limited.

4.4 Ablation Study

From the results of the ablation study presented in Table 2, we have the following observations. (1) The prototype framework of KPCL consistently outperforms its variants across all three datasets. This indicates that the combination of the RP and RA modules consistently leads to performance improvements, regardless of the distribution of the dataset. (2) The contribution of the RP and RA modules varies depending on the specific data distributions. Specifically, the RP module has a more significant impact on the Amazon-Book and Movieslens datasets compared to the RA module. This can be attributed to the presence of numerous invalid links within the KGs of these two datasets. However, the KG of Alibaba-iFashion contains numerous knowledge entities, but strictly limits the number of edges. In this case, the high-quality sparse knowledge data reduces the filtering effect of the RP module. Consequently, it can be deduced that the RP module structurally filters out potential noise in KG that does not contribute to recommendation tasks, while the RA module focuses on aligning features between knowledge entities and items.

5 RELATED WORK

5.1 Knowledge-aware Recommender Systems

Knowledge-aware Recommender Systems are a type of recommendation that leverages knowledge graphs to incorporate valuable information for item representation learning and user modeling. Generally, it can be broadly categorized as embedding-based, path-based, and GNN-based methods.

Embedding-based methods [14, 18, 20] utilize KG embedding techniques to learn entity embeddings as prior information of items. They subsequently utilize knowledge entity and relation embeddings to improve the semantic representations in recommender systems. For example, in [14], CKE integrates various types of side information into the collaborative filtering framework to construct a collaborative knowledge graph. The structural knowledge embeddings of items are encoded using TransR. However, these embedding-based methods cannot capture long-range semantics or sequential dependencies due to their disregard for high-order connectivity. This limitation restricts their representation ability for recommendation tasks.

Path-based methods [19, 21, 22, 57] have shown their ability to explore high-order connectivity by extracting paths from target user to item nodes via knowledge triplets. These paths are then employed to predict user preferences using recurrent neural networks or memory networks. An example of this is the utilization of Long Short-Term Memory (LSTM) in KPRN [19] to extract side information through meta-paths. However, path-based methods have inherent limitations. The effectiveness of recommendation heavily relies on the quality of meta-paths, which typically require labor-intensive feature engineering and identification of domain-specific patterns.

GNN-based Methods [15, 28, 31] leverage the information aggregation mechanism of GNN to learn representations from KG for recommendation tasks. KGAT [15], for example, builds a heterogeneous graph with user-item interactions and KG. It then applies an attention aggregation mechanism to propagate user and item embeddings on this graph. GNN-based methods have been widely adopted in recent research due to their ability to capture rich semantics representations and achieve state-of-the-art performance.

5.2 Self-Supervised Learning for Recommender Systems

Self-supervised learning, particularly contrastive learning, has gained increasing attention in the area of recommendation systems. By designing pretext tasks that provide additional supervised signals [52], contrastive learning aims to learn high-quality discriminative representations by distinguishing between supervised signals and noise [51]. Recent studies have shown that introducing contrastive learning improves the quality of graph embedding representation in terms of alignment and uniformity [39]. Many researchers have incorporated contrastive learning into their methods to construct self-supervised recommendation systems. For example, SGL [56] applies graph contrastive learning to graph recommendation systems by using random graph augmentation to generate contrastive views. Further, KGCL [37] introduces graph contrastive learning on KG to resist noise and address long-tail problems. It leverages additional signals from KG to enhance user-item representation learning for collaborative filtering. Recently, KGRec [39] applies two self-supervised learning methods, masked-encoder, and contrastive learning, to discover knowledge rationale and match the graph embeddings in cross-view. Different from the above existing self-supervised frameworks, our work KPCL proposes a novel preference contrastive learning paradigm from both macro and micro perspectives to align the intent from collaborative signals structurally.

6 CONCLUSION

In this paper, we proposed a novel preference contrastive learning method (KPCL) for recommendation. A self-supervised module driven by relation preference ranking and rationale attention was applied to align the collaborative signals. This method helps KG to filter out interfering noise and discard knowledge connects that contribute little to the recommendation task. Then, an augmented heterogeneous aggregator was conducted to optionally aggregate informative knowledge connections. Extensive experiments on three real-world datasets validated the superiority of KPCL over state-of-the-art baselines. This work provides valuable insight into aligning collaborative signals from the macro and micro perspectives. In the future, advanced methods of self-supervised preference learning for knowledge-aware recommender systems will be explored.

Acknowledgements

This work was supported by the National Natural Science Foundation of China [No. 72171183].

References

[1] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 1063–1073, 2023. 1

[2] Shang Liu, Zhenzhong Chen, Hongyi Liu, and Xinghai Hu. User-video co-attention network for personalized micro-video recommendation. In *The world wide web conference*, pages 3020–3026, 2019.

[3] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 world wide web conference*, pages 167–176, 2018. 1

[4] Yuntao Du, Jianxun Lian, Jing Yao, Xiting Wang, Mingqi Wu, Lu Chen, Yunjun Gao, and Xing Xie. Towards explainable collaborative filtering with taste clusters learning. In *Proceedings of the ACM Web Conference 2023*, pages 3712–3722, 2023. 1

[5] Lianghao Xia, Chao Huang, Jiao Shi, and Yong Xu. Graph-less collaborative filtering. In *Proceedings of the ACM Web Conference 2023*, pages 17–27, 2023.

[6] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*, pages 263–272. Ieee, 2008.

[7] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

[8] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008. 1

[9] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019. 1

[10] Yu Wang, Xin Xin, Zaiqiao Meng, Joemon M Jose, Fuli Feng, and Xiangnan He. Learning robust recommenders through cross-model agreement. In *Proceedings of the ACM Web Conference 2022*, pages 2015–2025, 2022.

[11] Zhendong Chu, Hongning Wang, Yun Xiao, Bo Long, and Lingfei Wu. Meta policy learning for cold-start conversational recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 222–230, 2023. 1

[12] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The world wide web conference*, pages 151–161, 2019. 1, 2

[13] Riku Togashi, Mayu Otani, and Shin'ichi Satoh. Alleviating cold-start problems in recommendation through pseudo-labelling over knowledge graph. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 931–939, 2021. 1

[14] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362, 2016. 1, 7, 9

[15] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019. 2, 3, 4, 7, 9

[16] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Exploring high-order user preference on the knowledge graph for recommender systems. *ACM Transactions on Information Systems (TOIS)*, 37(3):1–26, 2019.

[17] Xiaolong Liu, Liangwei Yang, Zhiwei Liu, Mingdai Yang, Chen Wang, Hao Peng, and Philip S Yu. Knowledge graph context-enhanced diversified recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 462–471, 2024. 1

[18] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9):137, 2018. 1, 9

[19] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5329–5336, 2019. 1, 9

[20] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*, pages 1835–1844, 2018. 1, 9

[21] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 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*, pages 1531–1540, 2018. 1, 9

[22] Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J Smola. An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 75–84, 2020. 1, 9

[23] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 285–294, 2019. 1

[24] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023. 1

[25] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[27] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. 1

[28] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 968–977, 2019. 1, 7, 9

[29] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. Ckan: Collaborative knowledge-aware attentive network for recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 219–228, 2020. 1

[30] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. Knowledge graph convolutional networks for recommender systems. In *The world wide web conference*, pages 3307–3313, 2019. 2

[31] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the web conference 2021*, pages 878–887, 2021. 2, 6, 7, 9

[32] Yunjun Gao, Yuntao Du, Yujia Hu, Lu Chen, Xinjun Zhu, Ziquan Fang, and Baihua Zheng. Self-guided learning to denoise for robust recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1412–1422, 2022. 2

[33] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 373–381, 2021.

[34] Xinjun Zhu, Yuntao Du, Yuren Mao, Lu Chen, Yujia Hu, and Yunjun Gao. Knowledge-refined denoising network for robust recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 362–371, 2023.

[35] Yu Tian, Yuhao Yang, Xudong Ren, Pengfei Wang, Fangzhao Wu, Qian Wang, and Chenliang Li. Joint knowledge pruning and recurrent graph convolution for news recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 51–60, 2021. 2

[36] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015. 2

[37] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. Knowledge graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1434–1443, 2022. 2, 5, 7, 9

[38] Hao Wang, Yao Xu, Cheng Yang, Chuan Shi, Xin Li, Ning Guo, and Zhiyuan Liu. Knowledge-adaptive contrastive learning for recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 535–543, 2023. 2, 6

[39] Yuhao Yang, Chao Huang, Lianghao Xia, and Chunzhen Huang. Knowledge graph self-supervised rationalization for recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3046–3056, 2023. 2, 3, 4, 5, 6, 7, 9

[40] Shuyao Wang, Yongduo Sui, Chao Wang, and Hui Xiong. Unleashing the power of knowledge graph for recommendation via invariant learning. In *Proceedings of the ACM on Web Conference 2024*, pages 3745–3755, 2024. 2, 3

[41] Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. Diffkg: Knowledge graph diffusion model for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 313–321, 2024. 2

[42] Gu Tang, Xiaoying Gan, Jinghe Wang, Bin Lu, Lyuwen Wu, Luoyi Fu, and Chenghu Zhou. Editkg: Editing knowledge graph for recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112–122, 2024. 2

[43] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. pages 452–461, 2009. 3, 5, 6

[44] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31 (2):357–370, 2018. 3

[45] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3

[46] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[47] Xiao Liu, Shiyu Zhao, Kai Su, Yukuo Cen, Jiezhong Qiu, Mengdi Zhang, Wei Wu, Yuxiao Dong, and Jie Tang. Mask and reason: Pre-training knowledge graph transformers for complex logical queries. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1120–1130, 2022. 4

[48] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2018. 4

[49] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020. 5, 6

Table 3: Statistics of the datasets.

Statistics	Movielens	Alibaba-iFashion	Amazon-book
Users	37,385	114,737	70,679
Items	6,182	30,040	24,915
Interactions	539,300	1,781,093	847,733
Entities	24,536	59,156	88,572
Relations	20	51	39
Triplets	237,155	279,155	2,557,746
Density	19.331	9.438	57.755

[50] Ding Zou, Wei Wei, Xian-Ling Mao, Ziyang Wang, Minghui Qiu, Feida Zhu, and Xin Cao. Multi-level cross-view contrastive learning for knowledge-aware recommender system. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1358–1368, 2022. 5, 7

[51] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 5, 9

[52] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 5, 9

[53] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. Pog: personalized outfit generation for fashion recommendation at alibaba ifashion. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2662–2670, 2019. 6

[54] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517, 2016. 6

[55] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017. 6

[56] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021. 7, 9

[57] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011. 9

A Appendix

A.1 Dataset Description

The statistics of datasets used in the experiments are reported in Table 3.

A.2 Impact of Aggregation Depth

In this subsection, the number of heterogeneous aggregation layers is explored. Here, we set the number of layers to be either 2 or 3 and summarized the results in Table 4. Since constructing rationale attention with a single relation set is not feasible, the case of KPCL-1 is excluded. Generally, longer paths with more stacked layers are expected to integrate information from longer-range connectivity into node representations. However, the results of Table 4 show that KPCL-2 outperforms KPCL-3 in most cases, except for the Amazon-book dataset where KPCL-3 performs slightly better in terms of

Table 4: Impact of the number of layers.

Layers	Movielens		Alibaba-iFashion		Amazon-book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
KPCL-2	**0.4647**	**0.3117**	**0.1181**	**0.0741**	**0.1609**	0.0859
KPCL-3	0.4617	0.3089	0.1165	0.0727	0.1605	**0.0866**

NDCG. Based on these results, we can conclude that: (1) For most datasets, two-layer aggregation is sufficient to extract necessary information for recommendation tasks. Adding more layers and exploring deeper information within KG may introduce additional noise, potentially making the model suboptimal. (2) In the more dense and complex KG like the case of Amazon-book, the exploration of high-order information introduces noise but also provides effective side information that benefits collaborative filtering.

A.3 Model Explainability Study

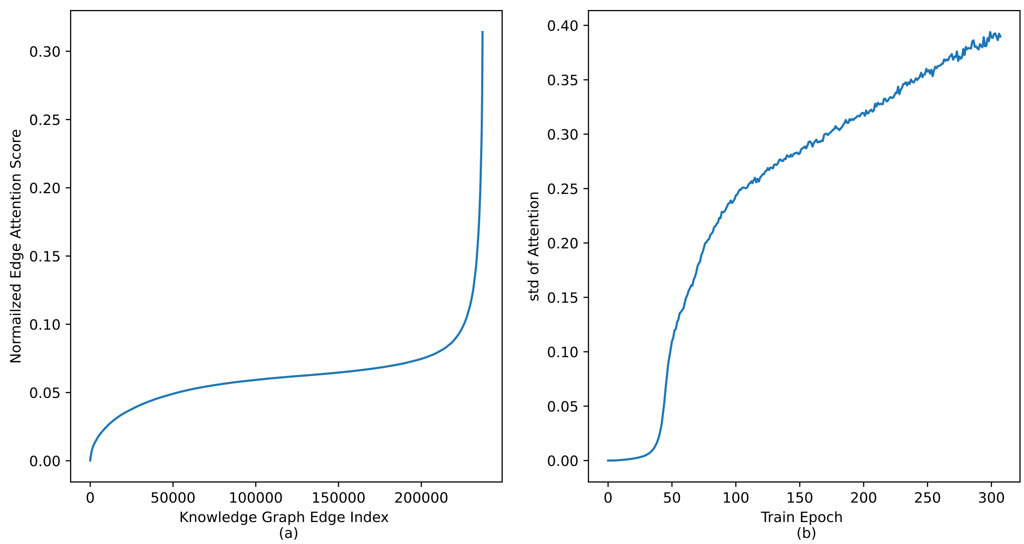

Figure 2: Explanation of KPCL training on Movielens dataset. The part (a) displays the attention score distribution of all edges after training. The part (b) visualizes the trend of standard deviation (std) of edge attention scores during the training process.

To analyze the preference contrastive learning method comprehensively, we record the distribution of attention scores and their dispersion. From the results reported in Figure 2 on sparse users, we have the following observations. Firstly, upon examining the sorted result of the normalized edge attention distribution, it is evident that only a very small proportion of knowledge edges achieve higher scores and produce a significant contribution in recommendation. Conversely, edges with scores close to zero are dropped optionally to avoid propagating noise. This indicates that the model focuses its attention on a subset of edges that are deemed more informative and relevant for making recommendations. Secondly, Part (b) of the figure demonstrates that the dispersion of edge attention, measured by the Standard Deviation (std), increases as the training steps progress. This suggests that the model effectively distinguishes between effective edges, which contribute significantly to recommendation tasks, and noise.

To provide an intuitive understanding of the explainability of our approach, we illustrate an example from the Movielens data in Figure 3. From this case, we can observe that KPCL carried out a path design process automatically based on the rationale score. For instance, let's consider entity 4940 as an example. In this case, even when there are multiple connections with the same type of relation, KPCL does not assign approximate weights blindly. Instead, it learns to assign weights based on the recommendation requirements. In this specific example, entity 4940 assigns the highest weight to entity 13140, rather than focusing on entity 17413, which has more connections. This highlights the

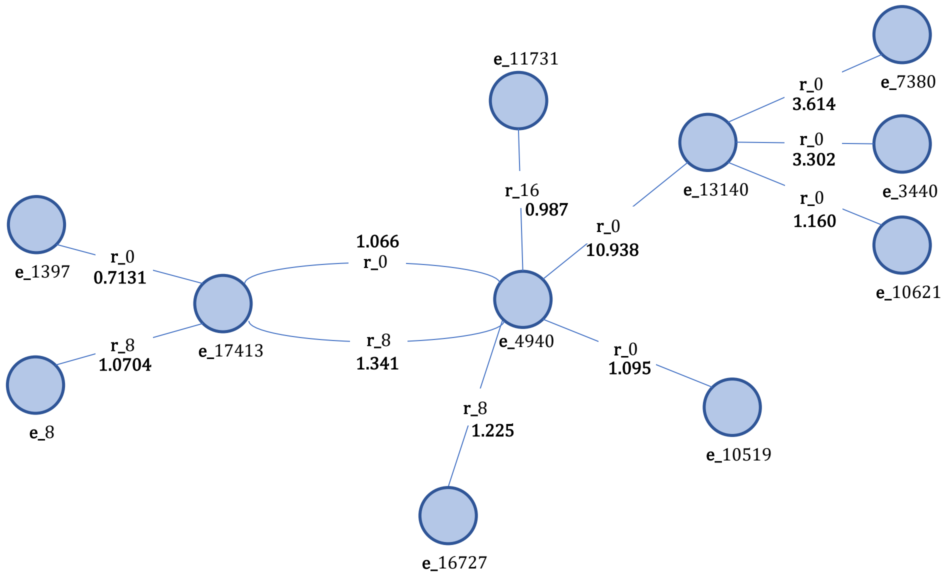

Figure 3: Explanation of a case of the rationale attentive mechanism in Movielens.

model's ability to learn the embeddings from a global and holistic perspective, taking into account the recommendation requirements and aligning with collaborative signals.