Knowledge-refined Denoising Network for Robust Recommendation

Xinjun Zhu
Zhejiang University
xjzhu@zju.edu.cn

Yuntao Du
Zhejiang University
ytdu@zju.edu.cn

Yuren Mao
Zhejiang University
yuren.mao@zju.edu.cn

Lu Chen
Zhejiang University
luchen@zju.edu.cn

Yujia Hu
Zhejiang University
charliehu@zju.edu.cn

Yunjun Gao
Zhejiang University
Alibaba-Zhejiang University Joint Institute of Frontier Technologies
gaoyj@zju.edu.cn

ABSTRACT

Knowledge graph (KG), which contains rich side information, becomes an essential part to boost the recommendation performance and improve its explainability. However, existing knowledge-aware recommendation methods directly perform information propagation on KG and user-item bipartite graph, ignoring the impacts of *task-irrelevant knowledge propagation* and *vulnerability to interaction noise*, which limits their performance. To solve these issues, we propose a robust knowledge-aware recommendation framework, called *Knowledge-refined Denoising Network* (KRDN), to prune the task-irrelevant knowledge associations and noisy implicit feedback simultaneously. KRDN consists of an adaptive knowledge refining strategy and a contrastive denoising mechanism, which are able to automatically distill high-quality KG triplets for aggregation and prune noisy implicit feedback respectively. Besides, we also design the self-adapted loss function and the gradient estimator for model optimization. The experimental results on three benchmark datasets demonstrate the effectiveness and robustness of KRDN over the state-of-the-art knowledge-aware methods like KGIN, MCCLK, and KGCL, and also outperform robust recommendation models like SGL and SimGCL. The implementations are available at https://github.com/xj-zhu98/KRDN.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommendation, Graph Neural Network, Knowledge Graph

ACM Reference Format:
Xinjun Zhu, Yuntao Du, Yuren Mao, Lu Chen, Yujia Hu, and Yunjun Gao. (*Corresponding author: Yuren Mao*). 2023. Knowledge-refined Denoising Network for Robust Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591707

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591707

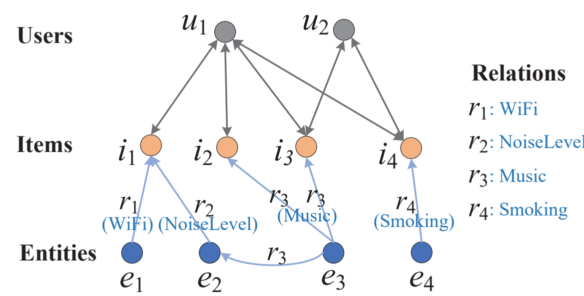


Figure 1: An example of knowledge-aware recommendation on Yelp2018 dataset.

1 INTRODUCTION

In the Internet era of information explosion, recommendation systems are widely deployed in real-life applications such as E-commerce, online advertisement, and social media platforms to provide personalized information services. Traditional recommendations (*e.g.,* collaborative filtering [7, 17, 23, 41]) heavily rely on historical user interaction data, which brings data sparsity and cold-start problems; furthermore, their vulnerability to interaction noise also degrades the recommendation performance [39, 44, 46]. Recently, knowledge graph (KG), which provides rich side information among items, has demonstrated great potential in alleviating cold-start issues and improving the robustness and explainability of recommendations.

Incorporating external knowledge from KG to learn high-quality user and item representations has become the concept of knowledge-aware recommendation. Early work [1, 36, 57] on this topic directly integrates knowledge graph embeddings with items to enhance representations. Then, to further improve the performance of knowledge-aware recommendation, knowledge meta-paths-based methods are proposed [16, 19, 43], which enriches the interactions with meta-paths from users to items for better exploiting user-item connectivities. However, due to the difficulty of obtaining informative meta-paths, these methods suffer from labor-intensive process [43], poor scalability [16], and unstable performance [51]. To address these issues, Graph Neural Networks (GNNs) [14, 22, 34] are adopted in knowledge-aware recommendation to achieve end-to-end recommendation by means of iteratively propagating high-order information over KG[33, 35, 37, 40, 42, 45, 53, 58]. These propagation-based methods can effectively gather multi-hop neighbors into representations, which enable to achieve the impressive performance of recommendation.

Despite effectiveness, we argue current propagation-based methods commonly have the following two limits:

- **Task-irrelevant Knowledge Propagation.** Previous research blindly aggregate all kinds of information in KG into item representations, regardless of their semantic relatedness with recommendation task. However, due to the scale and generalization of knowledge graphs [24], they can be rather noisy, and some facts in KG are semantically far away from recommendation scenarios. Taking the Figure 1 as an example, the three businesses i_1, i_2, i_3 interacted by u_1, where i_2, i_3 are resident in *music* (r_3), i_1 is associated with the *NoiseLevel* (r_2). It can be concluded that u_1 focuses on the ambience, while *WiFi* (r_1) is a more marginal attribute linked to i_1. Integrating these irrelevant facts such as (i_1, r_1, e_1) are not useful for learning high-quality user and item representations, and can also introduce unnecessary noise and thus degrade recommendation performance.
- **Vulnerable to Interaction Noise**. To align with the graph structure of KG, existing studies typically construct user-item interaction graph from implicit feedback and propagate collaborative information with GNNs. However, the recursive message passing scheme of GNNs is known to be vulnerable to the quality of the input graphs [4], and implicit feedback is inherently noisy [12, 39]. Directly performing propagation on such a noisy interaction graph would make the model difficult to learn users' real interests and degrades the performance. For example, u_1 and i_4 are related in structure, maybe because u_1 and u_2 have both been to i_3, the recommendation system recommends i_4 to u_1 according to the transient behavior similarity between the two users, and u_1 happens to have an interaction with i_4. However, from the knowledge information brought by KG, i_1, i_2, i_3 that u_1 interacts with are all environment-conscious businesses, while i_4 supports *smoking* (r_4), which is inconsistent with u_1's preference. So, u_1 is semantically distinct from i_4, which is probably a noisy interaction. Therefore, ignoring the noisy interactions would propagate misleading messages and contaminate the entire graph.

To tackle the aforementioned challenges, we carefully explore the conduction of knowledge and collaborative signals on KG and user-item graphs, and then propose a novel denoising scheme to prune the task-irrelevant knowledge and noisy interactions simultaneously. Specifically, we propose a new model, Knowledge-Refined Denoising Network (KRDN), which can not only make full use of relevant knowledge in KG to promote recommendation performance, but also show excellent robustness to noisy implicit interactions. KRDN consists of two components to correspondingly address the above problems:

- **Adaptive Knowledge Refining.** An adaptive pruning strategy is proposed to distill high-quality triplets and offer additional knowledge for the recommendation, which can be jointly optimized with downstream recommendation tasks. Besides, according to their relatedness with items, each KG triplet is recognized with a certain type of facts (*i.e.,* "item-item" facts, "item-attribute" facts, and "attribute-attribute" facts, detailed in Section 3.1.2). Based on pruned knowledge and multi-faceted facts, we design a novel compositional knowledge aggregation mechanism to effectively capture refined and multi-faceted contexts into representations for better characterizing items.
- **Contrastive Denoising Learning.** To avoid noisy message passing and improve the robustness of recommendation, we investigate the collaborative and knowledge similarities and devise a contrastive denoising mechanism to capture the divergence between them and identify noisy interactions for learning user true preference. Specifically, KRDN iteratively adjusts weights of possible noisy edges through a relation-aware self-enhancement mechanism in both soft and hard manners.

To this end, the newly proposed KRDN model is designed to i) adaptively refine knowledge associations, and ii) better capture users' true preferences by denoising interactions with the help of KG. We conduct extensive experiments on three real-world datasets to evaluate the performance of KRDN and existing methods. Experimental results show that our KRDN significantly outperforms all the start-of-the-art methods such KGIN [42], KGCL [53], MCCLK [58] and SimGCL [56]. Furthermore, KRDN is able to identify noisy edges and more robust than other methods.

In summary, our contributions are as follows:

- We approach knowledge-aware recommendation from a new perspective by refining knowledge associations and denoising implicit interactions simultaneously.
- This paper exploits the knowledge association in KG in a fine-grained way, which can not only learn to prune irrelevant triplets in the light of downstream supervision signals, but also aggregate multi-faceted facts compositionally for high-quality knowledge representations.
- We propose a contrastive denoising strategy by leveraging the semantic divergence between collaborative and knowledge aspects to better represent and propagate user true preference, which can greatly enhance the robustness of recommendations.
- Extensive experiments on three public benchmark datasets are conducted to demonstrate the superiority of KRDN.

2 PROBLEM FORMULATION

We begin by introducing structured data relating to our investigated problem, and then we formulate our task.

User-Item Bipartite Graph. In this paper, we concentrate on inferring the user preferences from the implicit feedback [31]. To be specific, the behavior data (*e.g.,* click, comment, purchase) involves a set of users $\mathcal{U} = \{u\}$ and items $\mathcal{I} = \{i\}$. We view user-item interactions as a bipartite graph $\mathcal{G}_b$, and construct the interaction matrix $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, where $|\mathcal{U}|$ and $|\mathcal{I}|$ denote the number of users and items, respectively. Each entry $\mathbf{R}_{ui} = 1$ if user u has interacted with item i, and $\mathbf{R}_{ui} = 0$ otherwise. Note that implicit feedback is inherently noisy [17, 31], and observed interactions are not necessarily positive [47, 54], which would lead to sub-optimal performance. We will discuss how to address the problem by leveraging complementary information of KG in Section 3.2.

Knowledge Graph. KGs hold structured data about real-world facts, like item attributes, concepts, or external commonsense. Let KG be a heterogeneous graph $\mathcal{G}_k = \{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where each triplet $(h, r, t) \in \mathcal{T}$ means that a relation r exists from head entity h to tail entity t; $\mathcal{T}$, $\mathcal{E}$ and $\mathcal{R}$ refer to the sets of triplets, entities, and relations in $\mathcal{G}_k$. For example, a triple (*Mark Hamill*, *ActorOf*, *Star War*) indicates that *Mark Hamill* is an *actor* of the movie *Star War*. Therefore, we can link items with entities ($\mathcal{I} \subset \mathcal{E}$) to offer auxiliary semantics to interaction data. However, observations in Figure 1 show that KG involves numerous task-irrelevant triplets, which cause a serious impact on recommendation. We will demonstrate how to model fine-grained facts over KG in Section 3.1.

Task Description. Given the user-item bipartite graph $\mathcal{G}_b$ and the KG $\mathcal{G}_k$, our task of knowledge-aware recommendation is to predict how likely that a user would adopt an item which he has not engaged with before.

3 METHODOLOGY

We present the proposed architecture of KRDN. Figure 2 shows the model framework, which consists of two key components: (1) adaptive knowledge refining, which uses parameterized binary masks to learn to remove irrelevant facts with an unbiased gradient estimator, meanwhile designs a compositional knowledge aggregator to effectively integrate different kinds of knowledge associations for contextual propagation; and (2) contrastive denoising learning, which focuses on the difference of collaborative and knowledge signals to identify noisy interactions in a contrastive way and iteratively performs relation-aware graph self-enhancement to augment user representations.

3.1 Adaptive Knowledge Refining

Unlike previous propagation-based methods [42, 53, 58] that directly integrate all kinds of information in KG into item representations, we aim to capture the most relevant knowledge associations which are beneficial for learning user preference. Specifically, we design an adaptive pruning mechanism that learns to prune extraneous facts with trainable stochastic binary masks, and devise a gradient estimator to jointly optimize them with the model.

Meanwhile, we argue that existing approaches are unable to characterize items properly because they do not differentiate item-related knowledge associations from the rest, and only aggregate KG information at a *coarse* granularity. Different entities in KG have different semantics for recommendation scenarios, and they play different roles in profiling items. This motivates us to perform a context-aware compositional aggregation mechanism to gather different semantics according to the relatedness of items.

3.1.1 ***Irrelevant Facts Pruning.*** As we discussed in Section 1, KG contains lots of noisy and task-irrelevant information, which is not useful or even degrades the performance. One straightforward solution is to manually construct a K-neighbor subgraph to constrain the receptive fields of nodes [10, 33], or randomly drop some edges to construct multi-view graph structure for contrastive learning [53, 58]. However, these approaches highly rely on the quality of graph construction, and cannot adaptively drop unnecessary edges according to the recommendation tasks. Hence, we turn to a parameterized method to jointly learn the optimal pruning strategy with downstream collaborative signals. Technically, we first attach each triplet in $\mathcal{T}$ with a binary mask $m \in \{0, 1\}$ to indicate whether the triplet should be dropped, so the post-pruned facts can be expressed as:

$$\tilde{\mathcal{T}} = \{(h_i, r_i, t_i) | m_i = 1\} \tag{1}$$

where $\tilde{\mathcal{T}}$ is the subset of $\mathcal{T}$. However, directly optimizing the masks M is computationally intractable due to its discrete, non-differentiability and combinatorial nature of $2^{|\mathcal{T}|}$ possible states [6, 18]. To address this challenge, we consider each m_i is subject to a Bernoulli distribution with parameter $\sigma(\alpha_i)$, so that

$$m_i \sim \mathrm{Bern}(\sigma(\alpha_i)) \tag{2}$$

where we choose the widely used sigmoid function as the deterministic function $\sigma(\cdot)$, so that the parameters α can be bounded with $(0, 1)$. To let masks M be jointly optimized with recommendation task, we integrate them with our target loss $\mathcal{L}$, and reformulate with Bernoulli parameters as:

$$\tilde{\mathcal{L}}(\alpha, \Theta) = \mathbb{E}_{M \sim \prod_{i=1}^{K} \mathrm{Bern}(m_i; \alpha_i)} [\mathcal{L}(M, \Theta)] \tag{3}$$

where $\mathbb{E}$ is the expectation, Θ denotes the rest parameters of models, and $\tilde{\mathcal{L}}$ is the evidence lower bound[1] (ELBO) [20] for objective $\mathcal{L}$ over the parameters α. To minimize the expected cost via gradient descent, we need to estimate the gradient $\nabla_\alpha \tilde{\mathcal{L}}(\alpha, \Theta)$. Note that there are several studies have been proposed to estimate the gradients for discrete variables, such as REINFORCE [48], Gumbel-Softmax [18], straight-through [2], hard concrete [26] and ARM [55]. However, those approaches either suffer from high variance or biased gradients. Thus, we adopt DisARM [6], a recently proposed unbiased and low-variance gradient estimator, to efficiently back-propagate the gradient of parameters α. We will introduce masks optimization in Section 3.4.2

3.1.2 ***Compositional Knowledge Aggregation.*** To better understand the semantic relatedness of KG triplets, we categorize $\mathcal{T}$ into three disjoint subsets $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$, in terms of their connectivities to items. Specifically, we denote $\mathcal{T}_1 = \{(h, r, t) | h, t \in \mathcal{I}\}$ as "item-item" facts, as both entities in these triplets are aligned with items. Similarly, $\mathcal{T}_2 = \{(h, r, t) | h \in \mathcal{I} \,\text{or}\, t \in \mathcal{I}\}$ stands for "item-attribute" facts, which means that one of the entities is related to item while the other acts as the attribute of it. The rest triplets $\mathcal{T}_3 = \{(h, r, t) | h, t \notin \mathcal{I}\}$ are "attribute-attribute" facts, where both entities are represented as attributes. As a result, we reorganize KG triplets with multi-facet facts, which can explicitly gather different knowledge associations. To denoise on message passing, we propose a new aggregation mechanism consisting of noisy message pruning and compositional knowledge aggregation. Specifically, we use $\mathcal{N}_h = \{(r, t) | (h, r, t) \in \mathcal{G}_k\}$ to represent the neighborhood entities and the first-order relations of item h in KG, and propose to integrate the multi-faceted relational context from neighborhood entities to generate the *knowledge representation* of entity h:

$$\mathbf{e}_h^{(1)} = \frac{1}{|\mathcal{N}_h|} \sum_{(r,t) \in \mathcal{N}_h} \mathrm{ReLU}\left(\mathbf{W}\phi(\mathbf{e}_t^{(0)}, \mathbf{e}_r)\right) \cdot m_{h,t}^{(0)} \tag{4}$$

where ReLU is the activation function, $m_{h,t}^{(0)} \in \{0, 1\}$ denotes whether triplet (h, r, t) should be pruned or not, and $\phi(\cdot)$ is the aggregation function which gathers information from neighboring entities and

[1]This can be derived by the Jensen's Inequality.

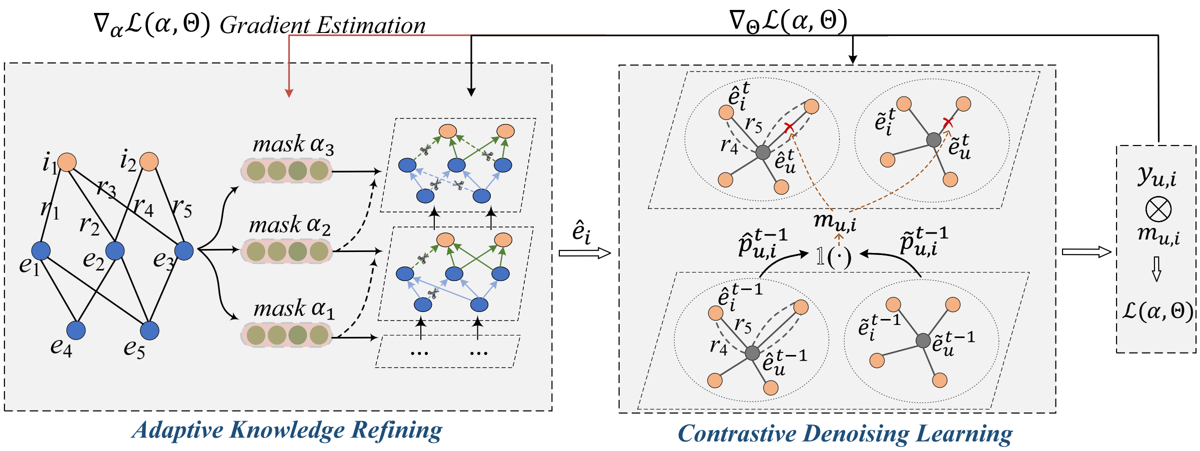


Figure 2: Illustration of the proposed KRDN framework. Interrupt task-irrelevant knowledge propagation in KG with multiple masks (left). Identify noise interactions according to the divergence between collaborative and knowledge signals (right).

corresponding relations. To differentiate different facts when aggregation, we design a compositional knowledge aggregator to integrate three kinds of semantics for avoiding interference between disparate information channels as follows:

$$\mathbf{W}\phi\left(\mathbf{e}_t, \mathbf{e}_r\right) = \begin{cases} \mathbf{W}_1\left(\mathbf{e}_t \odot \mathbf{e}_r\right), (h,r,t) \in \mathcal{T}_{1,3} \\ \mathbf{W}_2\left(\mathbf{e}_t \oplus \mathbf{e}_r\right), (h,r,t) \in \mathcal{T}_2 \end{cases} \tag{5}$$

where we use a relational-aware aggregation scheme [42] for same-level facts ("item-item" facts and "attribute-attribute" facts), and utilize additional operations for cross-level facts ("item-attribute" facts). Besides, we also introduce two trainable transformation matrices $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d\times d}$ to align the hidden semantics before aggregating the heterogeneous information together.

We stack more aggregation layers to explore the high-order knowledge associations for items. Technically, we recursively formulate the knowledge representations of item h after l layers as:

$$\mathbf{e}_h^{(l)} = \frac{1}{|\mathcal{N}_h|} \sum_{(r,t)\in\mathcal{N}_h} \mathrm{ReLU}\left(\mathbf{W}\phi(\mathbf{e}_t^{(l-1)}, \mathbf{e}_r)\right) \cdot m_{h,t}^{(l-1)} \tag{6}$$

3.2 Contrastive Denoising Learning

The second key component of KRDN framework is designed mainly to identify noisy interactions in user-item bipartite graph and propagate high-order discriminative collaborative signals to present user true preference. Existing methods [40, 42, 45] ignore the noise in interactions and directly aggregate all information from neighboring users/items, which would make the model difficult to differentiate between noise and user true preference and result in suboptimal user/item representations. More recently, some works [53, 58] focus on constructing different graph views and utilize contrastive learning to enhance the robustness of recommendation models. Unfortunately, this approach would inevitably lose the structure information and fail to identify fake interactions explicitly. Thus, we aim to leverage the divergence between collaborative signals and knowledge associations to filter the noisy interaction in an end-to-end manner. We illustrate our approach in Figure 2.

3.2.1 ***Denoising Collaborative Aggregation.*** The item representations generated from Section 3.1 contain refined knowledge associations, which are considered to have high confidence. By directly aggregating such item information from the interaction graph, we can obtain the knowledge representation of users and items. However, the user-item graph built upon implicit feedback inevitably contains noise [31]. Therefore, we initialize additional item representations to capture the pure collaborative signal individually as a comparison. To prune the noisy interaction, we focus on the divergence between collaborative and knowledge signals.

Since KRDN keeps the original graph structure instead of randomly perturbation [53], we are able to use the relative distance to assess the stability of i with respect to the importance of u as the basis for denoising:

$$m_{u,i} = \mathbb{1}\left(|\sigma(\tilde{p}_{u,i}) - \sigma(\hat{p}_{u,i})| < \gamma\right) \tag{7}$$

where $\mathbb{1}(\cdot)$ is a binary indicator function that returns 1 if the condition is true otherwise returns 0, and γ is a pre-defined threshold hyperparameter. $\tilde{p}_{u,i}$ and $\hat{p}_{u,i}$ respectively denote collaboration and knowledge similarities between u and i, and $\tilde{p}_{u,i}$ can be simply formulated as:

$$\tilde{p}_{u,i} = \frac{\exp\left(s\left(\tilde{\mathbf{e}}_u, \tilde{\mathbf{e}}_i\right)\right)}{\sum_{i'\in\mathcal{N}_{(u)}} \exp\left(s\left(\tilde{\mathbf{e}}_u, \tilde{\mathbf{e}}_{i'}\right)\right)} \tag{8}$$

where $\tilde{\mathbf{e}}_u$ and $\tilde{\mathbf{e}}_i$ are additional user and item representations, which are initialized to capture the pure collaborative signals, and $\mathcal{N}_{(u)}$ is used to represent the set of neighbors of node u in the user-item graph. $s(\cdot)$ denotes the inner product to estimate the similarity. As for $\hat{p}_{u,i}$, item representations from KG involving multi-relation semantics, and user preference mixing various information via message passing in the graph. Hence, we estimate the correlation degree between user u and item i across multiple relations. Each item has a relation set noted $\mathcal{R}_{(i)} = \{r|(h,r,t) \in \mathcal{T} \text{and } h \in \mathcal{I}\}$, and the similarity can be formulated as follows:

$$\hat{p}_{u,i} = \frac{\exp\left(\frac{1}{|\mathcal{R}_{(i)}|}\sum_{r\in\mathcal{R}_{(i)}} s\left(\mathbf{e}_r^\top \hat{\mathbf{e}}_u, \hat{\mathbf{e}}_i\right)\right)}{\sum_{i'\in\mathcal{N}_{(u)}} \exp\left(\frac{1}{|\mathcal{R}_{(i')}|}\sum_{r\in\mathcal{R}_{(i')}} s\left(\mathbf{e}_r^\top \hat{\mathbf{e}}_u, \hat{\mathbf{e}}_{i'}\right)\right)} \tag{9}$$

where $\hat{\mathbf{e}}_u$ and $\hat{\mathbf{e}}_i$ denotes knowledge-enhanced representation of u and i, and e_r denotes relation embedding. $\hat{p}_{u,i}$ embodies the personalized similarity of u for each interacted i according to the relations involved by i. Then the user preference can be acquired via the weighted sum of neighbors:

$$\hat{\mathbf{e}}_u = \hat{\mathbf{e}}_u + \sum_{i \in \mathcal{N}_{(u)}} m_{u,i} \hat{p}_{u,i} \hat{\mathbf{e}}_i \tag{10}$$

where $m_{u,i}$ and $\hat{p}_{u,i}$ refer to a combination of hard and soft ways to remove or reduce the weight of unreliable edges. $\tilde{\mathbf{e}}_u$ can be obtained in the same way.

3.2.2 ***Relation-aware Graph Self-enhancement.*** Due to the presence of noise, the aforementioned single-order denoising process is not sufficient to reduce the weight of the noisy edges and there is a chance of misclassification. Inspired by neighbor routing mechanism [28], we design a relation-aware self-enhancement mechanism to generate augmented representation and correlation degree between users and items in mentioned two kinds of signals, which is formulated as:

$$\hat{p}_{u,i} = \frac{\exp\left(\frac{1}{|\mathcal{R}_{(i)}|} \sum_{r \in \mathcal{R}_{(i)}} s\left(\mathbf{e}_r^\top \hat{\mathbf{e}}_u^{(n-1)}, \hat{\mathbf{e}}_i\right)\right)}{\sum_{i' \in \mathcal{N}_{(u)}} \exp\left(\frac{1}{|\mathcal{R}_{(i')}|} \sum_{r \in \mathcal{R}_{(i')}} s\left(\mathbf{e}_r^\top \hat{\mathbf{e}}_u^{(n-1)}, \hat{\mathbf{e}}_{i'}\right)\right)} \tag{11}$$

$$\hat{\mathbf{e}}_u^{(n)} = \frac{\hat{\mathbf{e}}_u^{(n-1)} + \sum_{i \in \mathcal{N}_{(u)}} m_{u,i} \hat{p}_{u,i} \hat{\mathbf{e}}_i}{\left\| \hat{\mathbf{e}}_u^{(n-1)} + \sum_{i \in \mathcal{N}_{(u)}} m_{u,i} \hat{p}_{u,i} \mathbf{e}_i \right\|_2} \tag{12}$$

where $m_{u,i}$ can be iteratively calculated by Eq. (7). With conducting Eq. (11) and (12) n times, noisy interactions will be gradually alienated and user representation $\hat{\mathbf{e}}_u^{(n)}$ is adjusted recursively towards the prototype of user preference [47]. Collaborative user representation $\tilde{\mathbf{e}}_u$ can be generated in the same way.

If $\hat{p}_{u,i}$ and $\tilde{p}_{u,i}$ are both small, the corresponding interaction has little impact on the formation of user preference and can be considered as a soft-style denoising manner, while if the divergence between $\hat{p}_{u,i}$ and $\tilde{p}_{u,i}$ is significant and exceeds the threshold, hard-style denoising is triggered.

3.3 Model Prediction

We obtain the two pair representation of item i and user u at separate layers in different signals after L layers, and then sum the multi-layer output as the final representation:

$$\hat{\mathbf{e}}_v = \sum\nolimits_{l=0}^{L} \hat{\mathbf{e}}_v^{(l)}, \qquad \tilde{\mathbf{e}}_v = \sum\nolimits_{l=0}^{L} \tilde{\mathbf{e}}_v^{(l)} \tag{13}$$

where subscript v denotes u or i. By doing so, we segregate the complementary information of collaborative and knowledge semantics in the final representations, and we use cosine similarity to forecast how likely the user u would engage with item i. Finally, the sum of the two-level similarity as the final prediction score $\hat{y}_{ui}$:

$$\hat{y}_{u,i} = \cos(\tilde{\mathbf{e}}_u, \tilde{\mathbf{e}}_i) + \cos(\hat{\mathbf{e}}_u, \hat{\mathbf{e}}_i) \tag{14}$$

3.4 Model Optimization

3.4.1 ***Self-adapting loss function.*** Benefiting from explicitly pruning low-confidence interactions, we build a similarity bank $\mathcal{M}$ to dynamically collect and adjust the weight of each "positive pair" from implicit feedback during the training process. In addition, to alleviate the convergence problem, we opt for the contrastive loss [29] that introduces more negative samples and penalizes uninformative ones to optimize KRDN:

$$\mathcal{L} = \sum_{u,i \in \mathcal{D}} \left[m_{u,i} \left(1 - \hat{y}_{u,i}\right)_+ + \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}} \left(\hat{y}_{u,j} - \beta\right)_+ \right] \tag{15}$$

where $m_{u,i} \in \mathcal{M}$ is the binary value derived from Section 3.2 to indicate whether each interaction (u, i) should be retained during the training process, thus preventing the generation of harmful gradients that interfere with the user's real preference. The goal is to maximize the similarity of positive pairs while decreasing the similarity of negative pairs with a margin β. Moreover, $\mathcal{D}$ is the interaction data, $\mathcal{N}$ is the negative item set through random sampling from unobserved items with user u, and $(\cdot)_+$ is the ramp function $max(\cdot, 0)$.

3.4.2 ***Indicators Gradient Estimation.*** An unbiased and low-variance gradient estimator DisARM [6] is adopted to efficiently calculate the gradient of parameters α. Let $M = (m_1, ..., m_K)$ be a vector of K independent Bernoulli variables with $m_i \sim \text{Bern}(\sigma(\alpha_i))$, and the gradient of $\tilde{\mathcal{L}}$ in Eq. 3 w.r.t α can be formulated as:

$$\nabla_\alpha \tilde{\mathcal{L}}(\alpha, \Theta) = \frac{1}{2} \sum_i (f(\mathbf{b}) - f(\tilde{\mathbf{b}}))((-1)^{\tilde{b}_i} \mathbb{1}_{b_i \neq \tilde{b}_i} \sigma(|\alpha_i|)) \tag{16}$$

where $(\mathbf{b}, \tilde{\mathbf{b}}) = ((b_1, \tilde{b}_1), ..., (b_K, \tilde{b}_K))^T$, discretized pair $(b_i, \tilde{b}_i) = (\mathbb{1}_{1-u_i<\sigma(\alpha_i)}, \mathbb{1}_{u_i<\sigma(\alpha_i)})$, and $u \sim U(0, 1)$ is sampled from Uniform distribution. $f(\mathbf{b})$ is the model loss obtained by setting each indicator m_i to 1 if $1 - u_i < \sigma(\alpha_i)$ in the forward pass of KRDN, 0 otherwise. The same strategy is applied to $f(\tilde{\mathbf{b}})$.

To this end, the gradient of binary indicators can be efficiently computed since: 1) Sampling from a Bernoulli distribution is replaced by sampling from a Uniform distribution between 0 and 1; 2) the estimator only involves two forward passes of model to calculate gradient, which can easily achieve with training.

3.5 Model Analysis

3.5.1 ***Model Size.*** The model parameters of KRDN consist of (1) trainable stochastic binary masks $\{\alpha_i | \alpha_i \in \alpha\}$; (2) ID embeddings of users, items, relations, and other KG entities $\{\hat{\mathbf{e}}_u, \hat{\mathbf{e}}_i, \tilde{\mathbf{e}}_u, \tilde{\mathbf{e}}_i, \mathbf{e}_e, \mathbf{e}_r | u \in \mathcal{U}, i \in \mathcal{I}, e \in \mathcal{E}, r \in \mathcal{R}\}$; and (3) two transformation parameters $W_{(1)}, W_{(2)}$ for compositional knowledge-refined aggregation.

3.5.2 ***Time Complexity.*** The time cost of KRDN mainly comes from two components: stochastic binary masks, aggregation, and self-enhancement schemes. The complexity of the stochastic binary masks are from DisARM, which requires two-forward pass of network and it's much less expensive than standard gradient back-propagation. In the aggregation over KG, $O(|\mathcal{G}_k|dL)$ is required to update entity representations, where $\mathcal{G}_k$, d, L denote the number of KG triplets, the embeddings size, and the number of layers. In the user-item graph aggregation and self-enhancement schemes, the computational complexity of user and item embeddings in dual information signals is $O(2|\mathcal{G}_b|dLn)$, where $\mathcal{G}_b$ denotes the number of interactions, and n is the iteration times. KRDN achieves comparable complexity to state-of-the-art knowledge-enhanced recommendation models.

Table 1: Statistics of the datasets.

		Alibaba-iFashion	Last-FM	Yelp2018
User-Item Interaction	#Users	114,737	23,566	45,919
	#Items	30,040	48,123	45,538
	#Interactions	1,781,093	3,034,796	1,185,068
Knowledge Graph	#Entities	59,156	58,266	90,961
	#Total-Triplets	279,155	464,567	1,853,704

4 EXPERIMENTS

We present empirical results to demonstrate the effectiveness of our proposed KRDN framework. The experiments are designed to answer the following three research questions:

- **RQ1:** How does KRDN perform, compared with the state-of-the-art knowledge-aware recommendation models and denoising recommendation models?
- **RQ2:** How do different components of KRDN (*i.e.,* adaptive knowledge refining, contrastive denoising learning, and the depth of propagation layers) and the noise in interactions and KG affect the performance of KRDN?
- **RQ3:** Can KRDN give intuitive impression of denoising results?

4.1 Experimental Settings

4.1.1 ***Dataset Description.*** We conduct experiments on three benchmark datasets: Alibaba-iFashion, Last-FM, and Yelp2018.

- **Alibaba-iFashion [5].** This is a fashion outfit dataset collected from Alibaba's online shopping system, which contains user-outfit click history, and outfits are viewed as items. Each outfit consists of several fashion staffs (*e.g.,* shoes, tops), and each staff has different fashion categories (*e.g.,* sweater, T-shirt).
- **Yelp2018[2].** This is a local business rating dataset collected by Yelp. We use the 2018 edition dataset of the Yelp challenge, where local businesses like restaurants and bars are viewed as the items.
- **Last-FM[3].** This is a music listening dataset collected from Last.fm music website, where the tracks are viewed as items. We take the subset of the dataset where the timestamp is from Jan 2015 to June 2015.

Following previous work [40, 42], we collect the two-hop neighbor entities of items in KG to construct the item knowledge graph for each dataset. Table 1 presents the overall statistics of the three datasets used in our experiments. Meanwhile, in order to evaluate the denoising capability of KRDN, we follow [32] to construct three corresponding polluted datasets by adding noise into three real-world datasets (denoted as Polluted Alibaba-iFashion, Polluted Last-FM, Polluted Yelp2018). Specifically, for each original dataset, we randomly drop the observed user-item interaction that $\mathbf{R}_{ui} = 1$ and sample an item that the user has not adopted before as noisy interaction to replace the dropped one. The rate of the replaced observed interactions is set to 5% by default, and we only inject noise in the training and validation sets.

4.1.2 ***Evaluation Metrics.*** We adopt the all-ranking strategy to evaluate the performance [42]. In the test set, we regard all the items that user has not interacted with before as negative samples. To evaluate the performance of top-N recommendation, we use adopt two widely-used evaluation metrics [40, 42] Recall@N and NDCG@N. We report the average results across all users in the test set with $N = 20$ by default.

[2]https://www.yelp.com/dataset
[3]https://grouplens.org/datasets/hetrec-2011

4.1.3 ***Baselines.*** For performance evaluation, We compare KRDN with various baselines, including KG-free (MF), embedding-based (CKE), propagation-based methods(KGNN-LS, KGAT, CKAN, KGIN, MCCLK, KGCL). Besides, we also include robust recommendation models (T-CE, SGL, SGCN, SimGCL) with pre-trained item representations from KG to demonstrate the denoising ability of KRDN.

- **MF** [31] is a benchmark factorization model, which only considers the user-item interactions and leaves KG untouched.
- **T-CE** [39] is a sample re-weighting method, which assigns zero or lower weight for large-loss samples on BCE loss to reduce the impact of noisy interaction.
- **SGL** [50] is a self-supervised method, which constructs multiple graph views and then conducts contrastive learning for robust learning. We adopt SGL with Edge Dropout (ED) strategy.
- **SGCN** [4] is a graph structure learning method, which applies a stochastic binary mask to prune noisy edges.
- **SimGCL** [56] is a state-of-the-art contrastive learning method, which proposes a simple contrastive strategy by adding uniform noises on embedding space to generate different views.
- **CKE** [57] utilizes TransR [24] to regularize item representations from KG, and feeds learned embeddings into MF framework.
- **KGNN-LS** [37] transforms KG into a user-specific graph and takes label smoothness into account.
- **KGAT** [40] combines the user-item graph with KG and recursively propagates the embeddings with an attention mechanism.
- **CKAN** [45] is based on KGNN-LS, which performs different aggregation mechanism on the user-item bipartite graph and KG respectively, to encode user and item representations.
- **KGIN** [42] models user-item interaction behaviors with different intents, and captures long-range semantics with a relation-aware aggregation scheme.
- **MCCLK** [58] is based on contrastive learning, which considers multi-level graph view, including structural, collaborative semantic views to mine additional supervised signal.
- **KGCL** [53] proposes a KG augmentation schema to guide a contrastive learning paradigm for robust recommendation.

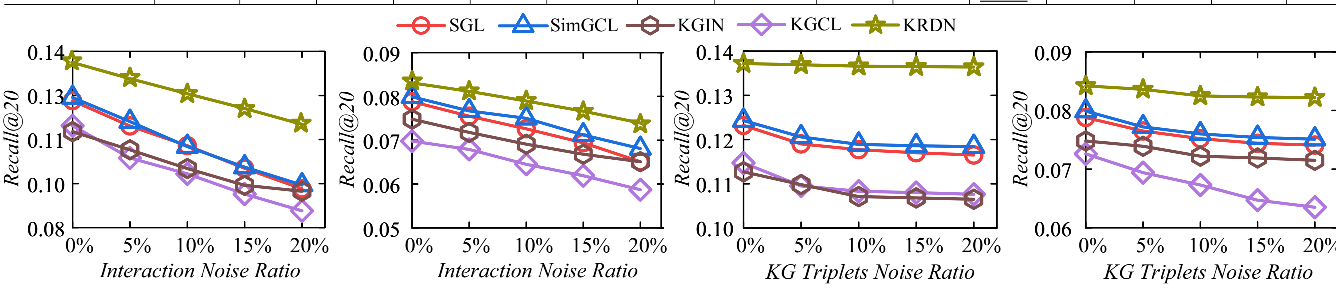
4.1.4 ***Parameter Settings.*** We implement KRDN in Pytorch, and have released our implementation to facilitate reproducibility. For a fair comparison, we fix the ID embedding size to 64, set the batch size to 4096, use the Xavier initializer [13] to initialize the model parameters, and optimize all models with Adam [21] optimizer. A grad search is applied for hyperparameters. We tune the learning rate among $\{10^{-4}, 10^{-3}, 10^{-2}\}$, the GNN layers L in $\{1, 2, 3\}$ and the pruning threshold γ among $\{0.1, ..., 0.5\}$. Besides, we set the number of negative samples $|\mathcal{N}|$ per user and the margin β of loss function to $\{200, 400, 400\}$ and $\{0.6, 0.7, 0.8\}$ for Alibaba-iFashion, Last-FM, and Yelp2018 datasets, respectively. Moreover, we carefully tune the other parameters for all baseline methods by following the original papers to achieve optimal performance.

4.2 Performance Comparison (RQ1)

We begin with the performance comparison *w.r.t.* Recall@20 and NDCG@20. The experimental results are reported in Table 2, where

Table 2: Overall performance comparison. "†" indicates the improvement of the KRDN over the baseline is significant at the level of 0.01. The highest scores are in Bold. R and N refer to Recall and NDCG, respectively.

Database	Method	MF	T-CE	SGCN	SGL	SimGCL	CKE	KGNN-LS	KGAT	CKAN	KGIN	MCCLK	KGCL	KRDN	%Imp.
Alibaba-iFashion	R@20	0.1095†	0.1093†	0.1145†	0.1232†	0.1243†	0.1103†	0.1039†	0.1030†	0.0970†	0.1147†	0.1089†	0.1127†	**0.1372**	10.38%
	N@20	0.0670†	0.0631†	0.0722†	0.0771†	0.0780†	0.0676†	0.0557†	0.0627†	0.0509†	0.0716†	0.0678†	0.0713†	**0.0879**	12.69%
Yelp2018	R@20	0.0627†	0.0705†	0.0768†	0.0788†	0.0799†	0.0653†	0.0671†	0.0705†	0.0646†	0.0698†	0.0696†	0.0748†	**0.0842**	5.38%
	N@20	0.0413†	0.0542†	0.0547†	0.0518†	0.0520†	0.0423†	0.0422†	0.0463†	0.0441†	0.0451†	0.0449†	0.0491†	**0.0544**	4.62%
Last-FM	R@20	0.0724†	0.0814†	0.0863†	0.0879†	0.0824†	0.0732†	0.0880†	0.0873†	0.0812†	0.0978†	0.0671†	0.0686†	**0.1023**	4.60%
	N@20	0.0617†	0.0683†	0.0759†	0.0775†	0.0736†	0.0630†	0.0642†	0.0744†	0.0660†	0.0848†	0.0603†	0.0629†	**0.0946**	11.56%
Polluted Alibaba-iFashion	R@20	0.0982†	0.0990†	0.1035†	0.1146†	0.1161†	0.0911†	0.0921†	0.0902†	0.0874†	0.1037†	0.0981†	0.1065†	**0.1312**	13.01%
	N@20	0.0607†	0.0584†	0.0639†	0.0714†	0.0722†	0.0630†	0.0471†	0.0542†	0.0448†	0.0643†	0.0613†	0.0672†	**0.0839**	16.20%
Polluted Yelp2018	R@20	0.0589†	0.0669†	0.0697†	0.0755†	0.0759†	0.0634†	0.0612†	0.0642†	0.0609†	0.0679†	0.0667†	0.0718†	**0.0816**	7.51%
	N@20	0.0392†	0.0477†	0.0480†	0.0492†	0.0495†	0.0412†	0.0401†	0.0407†	0.0416†	0.0436†	0.0422†	0.0472†	**0.0528**	6.67%
Polluted Last-FM	R@20	0.0711†	0.0807†	0.0858†	0.0879†	0.0948†	0.0849†	0.0863†	0.0845†	0.0805†	0.0960†	0.0668†	0.0731†	**0.1053**	9.69%
	N@20	0.0610†	0.0675†	0.0741†	0.0791†	0.0844†	0.0735†	0.0630†	0.0743†	0.0658†	0.0849†	0.0592†	0.0695†	**0.0988**	16.37%



(a) Alibaba-iFashion (b) Yelp2018 (c) Alibaba-iFashion (d) Yelp2018

Figure 3: Impact of different ratio of noise in user-item graph and knowledge graph.

Table 3: Impact of knowledge refining & denoising.

	Alibaba-iFashion		Yelp2018		Last-FM	
	recall	ndcg	recall	ndcg	recall	ndcg
w/o AKR	0.1317	0.0833	0.0826	0.0538	0.1008	0.0933
w/o CDL	0.1240	0.0794	0.0801	0.0521	0.0984	0.0905
w/o AKR&CDL	0.1225	0.0773	0.0789	0.0512	0.0974	0.0903

Table 4: Impact of the number of layers L.

	Alibaba-iFashion		Yelp2018		Last-FM	
	recall	ndcg	recall	ndcg	recall	ndcg
KRDN-1	0.1356	0.0866	0.0840	0.0544	0.1017	0.0939
KRDN-2	0.1365	0.0873	0.0842	0.0545	0.1023	0.0946
KRDN-3	0.1372	0.0879	0.0841	0.0545	0.1021	0.0941

Table 5: Impact of the iteration times n.

	Alibaba-iFashion		Yelp2018		Last-FM	
	recall	ndcg	recall	ndcg	recall	ndcg
n-1	0.1344	0.0852	0.0826	0.0527	0.1009	0.0929
n-2	0.1369	0.0875	0.0842	0.0545	0.1023	0.0946
n-3	0.1372	0.0879	0.0841	0.0543	0.1023	0.0945

%Imp. denotes the relative improvement of the best performing method (starred) over the strongest baselines (underlined). We have the following observations:

- KRDN consistently yields the best performance on all the datasets. In particular, it achieves significant improvement even the strongest baselines *w.r.t.* NDCG@20 by 12.69%, 4.62%, and 11.56% in Alibaba-iFashion, Yelp2018, and Last-FM, respectively. We attribute these improvements to the fine-grained modeling and denoising collaborative learning of KRDN: (1) By pruning irrelevant facts from KG with parameterized masks and introducing multi-channel information aggregation scheme, KRDN is able to explicitly capture important triplets for recommendation and aggregating different knowledge associations. In contrast, all baselines rely on a single aggregation scheme and ignore the different contributions of various facts in KG. (2) Benefited from the contrastive denoising scheme, KRDN can explicitly prune noisy implicit interactions by contrasting semantic distance between collaborative and knowledge aspects, while other baselines fail to leverage additional knowledge as denoising signals.
- When the extra noise is injected into the training data, KRDN also yields significant improvement. Compared with the strongest baselines, KRDN achieves around 12% performance improvement on three polluted datasets. For advanced knowledge-aware methods (*e.g.,* KGIN and KGCL), the injected noise significantly decreases their performances, while robust recommenders (*e.g.,* SGCN and SGL) are relatively less affected than their corresponding base models since they incorporate the denoising mechanism. Meanwhile, it is worth noting that most propagation-based methods are more sensitive to noise compared to MF and embedding-based methods. The reason is that message-passing scheme in GNN enlarges the negative impact of noise. KRDN removes irrelevant knowledge neighbors via applying learnable masks and prunes noisy interactions by contrasting collaborative and knowledge semantics, which can enhance the robustness and achieve better performances than other baselines.
- Although the side information of KG is important to improve the explainability and accuracy of recommendations, we also find that existing robust recommendation models (*e.g.,* SGL and SimGCL) show competitive or even better performance compared with knowledge-aware recommendations (*e.g.,* KGIN and KGCL). One possible reason is that current knowledge-aware methods fail to fully explore the power of KG in a fine-grained manner,

while our model is able to concentrate on clean triplets which are most useful for recommendation.

4.3 Study of KRDN (RQ2)

As knowledge refining and denoising are at the core of KRDN, we conduct ablation studies to investigate the effectiveness. Specifically, how the presence of noisy interactions and facts, the adaptive knowledge refining, the contrastive denoising learning, the iteration times, and the number of propagation layers affect our model.

4.3.1 ***Robustness to Noisy Interactions and Facts.*** We first conduct experiments to evaluate the robustness of KRDN with different ratios of noise. Following the dataset construction process described in Section 4.1.1, we pollute the training set and the validation set by replacing a certain ratio of original interactions with random noisy interactions, while keeping the testing set unchanged. Besides, we also add noise to the KG by randomly dropping tail entities t and selecting new tail entities t' for these triplets.

Figure 3(a) and figure 3(b) show the experimental results (Recall@20) on polluted Aliababa-iFashion and Yelp2018 dataset. From the figure, we can observe that increasing the ratio of noisy interactions significantly reduces the performances of all baseline methods. The performance degradation of KRDN is much smaller than that of other methods, especially on Yelp2018 dataset. And the performance of KRDN is consistently better than baseline methods. The gap becomes larger when the ratio of noises increases from 0% to 20%. These observations further confirm the importance of denoising interactions in recommendation and demonstrate the robustness and effectiveness of KRDN.

In addition, figure 3(c) and figure 3(d) demonstrate the recommendation results on two datasets with noisy KG triplets. We find that KRDN is much more robust compared with other baselines, which remains nearly the same performance when the noisy triplets increase. However, other methods like KGCL show dramatic performance degradation, since they cannot differentiate the contribution of different facts in KG and also fail to prune irrelevant knowledge for recommendation tasks.

4.3.2 ***Impact of knowledge refining & denoising.*** We then verify the effectiveness of adaptive knowledge refining and contrastive denoising learning. To this end, three variants of KRDN are constructed by (1) removing the adaptive knowledge refining and contrastive collaborative denoising, called KRDN $_{\text{w/o AKR\&CDL}}$, (2) replacing the adaptive knowledge refining with simple single facet aggregation, termed as KRDN $_{\text{w/o AKR}}$, and (3) discarding the contrastive denoising learning, named KRDN $_{\text{w/o CDL}}$. We summarize the results in Table 3.

Compared with the complete model of KRDN in Table 2, the absence of the adaptive knowledge refining and contrastive denoising learning dramatically degrades the performance, indicating the necessity of fine-grained KG modeling and denoising collaborative signals. Specifically, KRDN $_{\text{w/o AKR\&CDL}}$ directly aggregates all knowledge associations and ignores the noise in both KG and interactions, and thus, it fails to profile items properly and propagate information for learning use. Analogously, leaving the fine-grained knowledge associations unexplored (*i.e.,* KRDN $_{\text{w/o AKR}}$) also downgrades the performance. Although KRDN $_{\text{w/o CDL}}$ retains the fine-grained knowledge modeling for characterizing items, it is unable to provide discriminative signals for identifying user real behavior, incurring suboptimal user preference modeling.

4.3.3 ***Impact of Model Depth.*** We also explore the impact of the number of aggregation layers. Stacking more layers is able to collect the high-order collaborative signals and knowledge associations for better capturing of the latent user behavior patterns but at a higher cost. Here, we search L in the range of $\{1, 2, 3\}$, and report the results in Table 4. We have the following observations:

- Generally speaking, increasing the aggregation layers can enhance the performance, especially for Alibaba-iFashion datasets. We attribute such improvement to two reasons: (1) Gathering more relevant collaborative signals and knowledge association could provide informative semantics for learning high-quality representations, deepening the understanding of user interest. (2) The denoising module explicitly encodes both items' profiles from KG and users' behaviors from interactions, which fully explores the power of KG for robust preference learning.
- It is worth mentioning that our model is less sensitive to the model depth, compared with other propagation-based methods [40, 42, 53]. Specifically, KRDN could achieve competitive performance even when $L = 1$. This is because our fine-grained knowledge refining and denoising schemes can directly capture the most useful patterns from both user-item interactions and KG, while other methods need more layers to encode the latent semantics from the mixed and obscure information.

4.3.4 ***Impact of Self-enhancement Iteration Times.*** To evaluate the effect of self-enhancement on contrastive denoising learning, we design experiments under different times of iteration. We search n in the range of $\{1, 2, 3\}$, and report the result in Table 5.

The performance of the model improves when the number of iterations is increased from 1 to 2. It indicates that iteratively executing the relation-aware graph self-enhancement is able to provide superior user profiles via adjusting the user representations towards the prototype of user preference and gradually reducing the weight of noisy neighbors. When we continue to increase the number of iterations, the model performance does not change much, which is probably due to the fact that we perform self-enhancement operations in each GNN layer, making the representations smooth.

4.4 Case Study (RQ3)

In this section, we present an example to provide an intuitive impression of KRDN's explainability. Toward that, we randomly select a user u_{45716} from Yelp2018, and four directly connected items and their associated entities. As shown in Figure 4, we find that:

- Benefiting from knowledge refining in the knowledge graph with stochastic binary masks, we can intuitively infer the reason for some triplets that are regarded as task-irrelevant facts and are pruned. For example, the collaborative knowledge graph in Figure 4 shows that the user prefers these businesses with *WiFi* (r_6) and *GoodForKids* (r_{21}), and the triplet $(i_{23879}, r_{29}, e_{45704})$ which contains the attribute of *WheelchairAccessible* (r_{29}) is irrelevant to the user profile. Meanwhile, the model assigns the triplet $(i_{23879}, r_{29}, e_{45704})$ a lower probability value of 0.0841, so that it

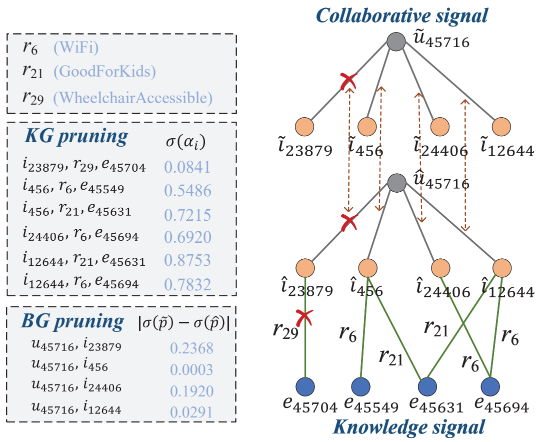


Figure 4: Explainability of denoising in Yelp2018. BG denotes a bipartite graph.

has a small chance of being selected and avoids affecting the modeling of the user profile.

- The foundation for pruning noise interactions can be clearly perceived. u_{45716} and i_{23879} interact with each other and thus have collaborative similarity, but i_{23879} contains attributes (e.g., "WheelchairAccessible") that are far from u_{45716}'s preferences. Especially, Figure 4 shows that the similarity of (u_{45716}, i_{23879}) has a great disagreement between collaborative and knowledge signals, so this interaction could be pruned when the threshold is set to 0.2.

5 RELATED WORK

5.1 Knowledge-aware Recommendation

Existing Knowledge-aware recommendation methods can be roughly grouped into three categories: embedding-based [1, 3, 36, 57], path-based [16, 19, 43] and propagation-based methods [8, 9, 35, 37, 38, 40, 42, 45, 53, 58]. **Embedding-based methods** learn entities and relations embeddings in KG via knowledge graph embedding (KGE) methods (*e.g.,* TransR [24]) to strengthen the semantic representation in recommendation. For example, CKE [57] utilizes TransR to learn the knowledge representation of items, and incorporates learned embeddings into matrix factorization (MF) [31]. Although these methods reveal simplicity and flexibility via KGE, they fail to capture the long-range dependence of user-item relations. **Path-based methods** explore the long-range connectivity among users and items by constructing different semantic paths via KG. Those paths are used to predict user profiles with recurrent neural networks [43] or attention mechanism [16]. For instance, KPRN [43] extracts the meta-paths via KG entities and relations to model high-order relations of user-item interactions by RNNs. However, defining proper meta-paths is time-consuming for complicated knowledge graphs and inevitably leads to poor generalization for different recommendation scenarios [19, 27]. **Propagation-based methods** are inspired by the information aggregation mechanism of graph neural network (GNNs) [14, 15, 22, 34, 41, 49], which iteratively integrate multi-hop neighbors into node representation to discover high-order connectivity. For instance, KGAT [40] constructs a collaborative knowledge graph (CKG) using user-item interactions and KG, then performs an attentive aggregation mechanism on it. KGIN [42] integrates long-range semantics of relation paths by a new aggregation scheme and disentangles user preference behind user-item interactions by utilizing auxiliary knowledge for better interpretability. Most recently, MCCLK [58] and KGCL [53] combine a contrastive learning paradigm and build cross-view contrastive frameworks as additional self-discrimination supervision signals to enhance robustness. However, most of them fail to consider the negative impacts of task-irrelevant triplets in KG. Our work can prune task-irrelevant knowledge associations and noisy implicit feedback simultaneously.

5.2 Denoising Recommender Systems

Considerable attention has been paid to the robustness of recommendation systems. Especially, implicit feedback could be vulnerable to inevitable noise and then degrade the recommendation performance [12, 32, 39, 44, 46]. Some work has gone into dealing with the noisy implicit feedback problem. Sample selection is a simple idea, which selects informative samples and then trains the model with them [11, 46]. For instance, WBPR [11] assigns different sampling probabilities according to item prevalence. IR [46] discovers noisy samples based on the difference between predictions and labels. Moreover, sample re-weighting is a valid class of methods [39, 44]. For example, T-CE [39] considers that noisy examples would have larger loss values, and hence assigns lower weights to high-loss samples. Besides, some recent studies use auxiliary information [52] or design model-specific structures [4, 50, 54] to achieve denoising. For instance, DFN [52] uses additional explicit feedback (*e.g.,* like and dislike) to extract clean information from noisy feedback. SGCN [4] explicitly prunes the irrelevant neighbors in the message-passing stage through sparsity and low-rank constraints. Most recently, [25, 30, 50, 56] utilize contrastive learning as auxiliary supervision signals. For example, SimGCL [56] constructs augmented views by adding uniform noise. However, little effort has been done toward performing explicit denoising by utilizing knowledge graphs.

6 CONCLUSION

In this paper, we proposed a new knowledge-aware robust recommendation method KRDN, which solves noise issues in KG and user-item bipartite graphs simultaneously. KRDN can eliminate irrelevant semantics in the knowledge graph and reduce the interference of user history noise interaction (*e.g.,* wrong click, wrong purchase), so as to achieve more satisfactory personalized recommendations with excellent interpretability in various real-world scenarios. Extensive experiments on three real-world datasets have demonstrated the superiority of KRDN. In the future, we plan to investigate the dynamics of noise, because users' interests evolve over time and so do the patterns of noise, thus combining user interaction sequences and temporal knowledge graphs to be able to locate noise behavior more precisely.

ACKNOWLEDGMENTS

This work was supported by the NSFC under Grants No. (62025206, 61972338, and 62102351). Yuren Mao is the corresponding author of the work.

REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation. *Algorithms* 11, 9 (2018), 137–146.

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* (2013).

[3] Yixin Cao, Xiang Wang, Xiangnan He, Zikun hu, and Tat-Seng Chua. 2019. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences. In *WWW*. 151–161.

[4] Huiyuan Chen, Lan Wang, Yusan Lin, Chin-Chia Michael Yeh, Fei Wang, and Hao Yang. 2021. Structured Graph Convolutional Networks with Stochastic Masks for Recommender Systems. In *SIGIR*. 614–623.

[5] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. 2019. POG: personalized outfit generation for fashion recommendation at Alibaba iFashion. In *KDD*. 2662–2670.

[6] Zhe Dong, Andriy Mnih, and George Tucker. 2020. DisARM: An Antithetic Gradient Estimator for Binary Latent Variables. In *NeurIPS*. 18637–18647.

[7] Yuntao Du, Jianxun Lian, Jing Yao, Xiting Wang, Mingqi Wu, Lu Chen, Yunjun Gao, and Xing Xie. 2023. Towards Explainable Collaborative Filtering with Taste Clusters Learning. In *WWW*.

[8] Yuntao Du, Xinjun Zhu, Lu Chen, Ziquan Fang, and Yunjun Gao. 2022. MetaKG: Meta-learning on Knowledge Graph for Cold-start Recommendation. *TKDE* (2022).

[9] Yuntao Du, Xinjun Zhu, Lu Chen, Baihua Zheng, and Yunjun Gao. 2022. HAKG: Hierarchy-Aware Knowledge Gated Network for Recommendation. In *SIGIR*. 1390–1400.

[10] Yufei Feng, Binbin Hu, Fuyu Lv, Qingwen Liu, Zhiqiang Zhang, and Wenwu Ou. 2020. ATBRG: Adaptive Target-Behavior Relational Graph Network for Effective Recommendation. In *SIGIR*. 2231–2240.

[11] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2012. Personalized ranking for non-uniformly sampled items. In *Proceedings of KDD Cup 2011*. PMLR, 231–247.

[12] Yunjun Gao, Yuntao Du, Yujia Hu, Lu Chen, Xinjun Zhu, Ziquan Fang, and Baihua Zheng. 2022. Self-Guided Learning to Denoise for Robust Recommendation. In *SIGIR*. 1412–1422.

[13] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*.

[14] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*. 1025–1035.

[15] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. 639–648.

[16] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging Meta-Path Based Context for Top- N Recommendation with A Neural Co-Attention Model. In *KDD*. 1531–1540.

[17] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*. 263–272.

[18] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR*.

[19] Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J. Smola. 2020. An Efficient Neighborhood-Based Interaction Model for Recommendation on Heterogeneous Graph. In *KDD*. 75–84.

[20] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. 1999. An Introduction to Variational Methods for Graphical Models. *Machine learning* 2, 37 (1999), 183–233.

[21] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *ICLR*.

[22] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[23] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *KDD*. 426–434.

[24] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*. 2181–2187.

[25] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *WWW*. 2320–2329.

[26] Christos Louizos, Max Welling, and Diederik P Kingma. 2017. Learning sparse neural networks through L_0 regularization. In *ICLR*.

[27] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-Learning on Heterogeneous Information Networks for Cold-Start Recommendation. In *KDD*. 1563–1573.

[28] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled graph convolutional networks. In *ICML*. PMLR, 4212–4221.

[29] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A Simple and Strong Baseline for Collaborative Filtering. In *CIKM*. 1243–1252.

[30] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The world is binary: Contrastive learning for denoising next basket recommendation. In *SIGIR*. 859–868.

[31] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. 452–461.

[32] Changxin Tian, Yuexiang Xie, Yaliang Li, Nan Yang, and Wayne Xin Zhao. 2022. Learning to Denoise Unreliable Interactions for Graph Collaborative Filtering. In *SIGIR*. 122–132.

[33] Ke Tu, Peng Cui, Daixin Wang, Zhiqiang Zhang, Jun Zhou, Yuan Qi, and Wenwu Zhu. 2021. Conditional Graph Attention Networks for Distilling and Refining Knowledge Graphs in Recommendation. In *CIKM*. 1834–1843.

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[35] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *CIKM*. 417–426.

[36] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *WWW*. 1835–1844.

[37] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-Aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In *KDD*. 968–977.

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

[39] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *WSDM*. 373–381.

[40] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *KDD*. 950–958.

[41] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*. 165–174.

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

[43] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable Reasoning over Knowledge Graphs for Recommendation. In *AAAI*. 5329–5336.

[44] Yu Wang, Xin Xin, Zaiqiao Meng, Joemon M Jose, Fuli Feng, and Xiangnan He. 2022. Learning Robust Recommenders through Cross-Model Agreement. In *WWW*. 2015–2025.

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

[46] Zitai Wang, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2021. Implicit Feedbacks are Not Always Favorable: Iterative Relabeled One-Class Collaborative Filtering against Noisy Interactions. In *MM*. 3070–3078.

[47] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *Proceedings of the 28th ACM international conference on multimedia*. 3541–3549.

[48] Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine learning* 8, 3–4 (1992), 229–256.

[49] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying Graph Convolutional Networks. In *ICML*. 6861–6871.

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

[51] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *SIGIR*. 285–294.

[52] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2020. Deep Feedback Network for Recommendation.. In *IJCAI*. 2519–2525.

[53] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *SIGIR*. 1434–1443.

[54] Yonghui Yang, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2021. Enhanced Graph Learning for Collaborative Filtering via Mutual Information Maximization. In *SIGIR*. 71–80.

[55] Mingzhang Yin and Mingyuan Zhou. 2019. ARM: Augment-REINFORCE-Merge Gradient for Stochastic Binary Networks. In *ICLR*.

[56] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *SIGIR*. 1294–1303.

[57] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative Knowledge Base Embedding for Recommender Systems. In *KDD*. 353–362.

[58] Ding Zou, Wei Wei, Xian-Ling Mao, Ziyang Wang, Minghui Qiu, Feida Zhu, and Xin Cao. 2022. Multi-level cross-view contrastive learning for knowledge-aware recommender system. In *SIGIR*. 1358–1368.