# Knowledge Graph Self-Supervised Rationalization for Recommendation

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

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

Lianghao Xia
University of Hong Kong
aka_xia@foxmail.com

Chunzhen Huang
Wechat, Tencent
chunzhuang@tencent.com



## ABSTRACT

In this paper, we introduce a new self-supervised rationalization method, called KGRec, for knowledge-aware recommender systems. To effectively identify informative knowledge connections, we propose an attentive knowledge rationalization mechanism that generates rational scores for knowledge triplets. With these scores, KGRec integrates generative and contrastive self-supervised tasks for recommendation through rational masking. To highlight rationales in the knowledge graph, we design a novel generative task in the form of masking-reconstructing. By masking important knowledge with high rational scores, KGRec is trained to rebuild and highlight useful knowledge connections that serve as rationales. To further rationalize the effect of collaborative interactions on knowledge graph learning, we introduce a contrastive learning task that aligns signals from knowledge and user-item interaction views. To ensure noise-resistant contrasting, potential noisy edges in both graphs judged by the rational scores are masked. Extensive experiments on three real-world datasets demonstrate that KGRec outperforms state-of-the-art methods. We also provide the implementation codes for our approach at https://github.com/HKUDS/KGRec.

## CCS CONCEPTS

• **Information systems** → **Recommender systems**.

## KEYWORDS

Recommendation, Self-Supervised Learning, Knowledge Graph

**ACM Reference Format:**
Yuhao Yang, Chao Huang, Lianghao Xia, and Chunzhen Huang. 2023. Knowledge Graph Self-Supervised Rationalization for Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, Austin, TX, USA, 11 pages. https://doi.org/10.1145/3580305.3599400

*Chao Huang is the corresponding author.


*KDD '23, August 6–10, 2023, Long Beach, CA, USA*

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599400

## 1 INTRODUCTION

With the rise of information overload, recommender systems have become a critical tool to help users discover relevant items of interest [24, 43]. Among the leading paradigms in this field is collaborative filtering (CF), which assumes that users with similar interactions share similar interests in items [9, 19, 41]. CF has proven to be effective in a wide range of applications and has driven significant advances in the field of recommender systems.

In recent years, collaborative filtering (CF) frameworks have undergone significant improvements with the introduction of neural networks and latent embedding for users and items, leading to effective enhancements for traditional matrix factorization methods (*e.g.,* [7, 9, 15]). Moreover, novel models that integrate variational autoencoders, attention mechanisms, and graph neural networks have further increased the performance of CF (*e.g.,* [3, 8, 19, 34]). However, the sparsity of user-item interactions fundamentally limits the scope of performance improvement. To address this issue, incorporating a knowledge graph (KG) as a rich information network for items has gained traction in collaborative filtering, leading to knowledge graph-enhanced recommendation.

The exploration of knowledge graph-enhanced recommendation begins with embedding-based methods and path-based methods. Specifically, some studies [2, 27, 49] incorporate transition-based knowledge graph embedding, such as TransR [20], into item embedding to enrich user and item modeling. Other studies [36, 48] focus on extracting semantically meaningful meta-paths from the KG and perform complex modeling of users and items along these meta-paths. To unify embedding-based and path-based methods in a mutually beneficial manner, recent research has adopted powerful graph neural networks (GNNs) to capture multi-hop high-order information through propagation and aggregation on the KG. These state-of-the-art solutions include [30, 33, 35].

Although knowledge graphs have proven effective for improving recommendation systems, they can also introduce noise and sparsity issues, leading to sub-optimal performances [26]. To address these issues, recent studies propose using contrastive learning (CL) for better knowledge-aware recommendation. For example, KGCL [44] applies stochastic graph augmentation on the KG and performs CL to address noisy entity and long-tail problems in the KG. [52] design a cross-view CL paradigm between the KG and user-item graph to improve KG representation learning with real labels from recommendation. However, we argue that these methods adopt either simple random augmentation or intuitive cross-view

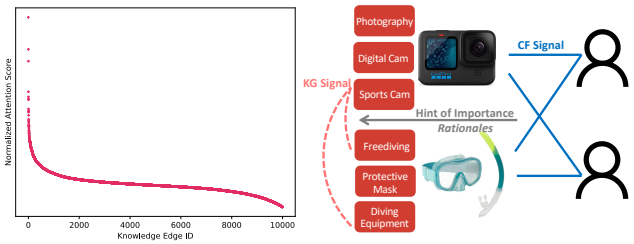



**Figure 1: The left figure displays a distribution of attentive scores for knowledge triplets in the baseline method of KGAT, which is skewed towards the tail end. On the other hand, the right figure suggests that we can determine the rationality of knowledge triplets for recommendation by analyzing the training labels of user-item interactions.**

information, failing to consider the important latent rationales between the KG and recommendation task.

Figure 1 presents the distribution of attention scores of knowledge triplets in KGAT on the left, and a motivating case on the right that illustrates the rationales in the KG emphasized by CF signals. The distribution of attention scores in the KGAT model shows that only a small proportion of knowledge triplets have high attention scores and are thus highly contributive to recommendation as rationales. The remaining knowledge triplets exhibit a long tail of low scores in the distribution and are less informative in the network. To better understand the relationship between KG and CF signals, we provide an example of an e-commerce platform where users often purchase diving glasses and underwater cameras together. To make accurate predictions, the connections with common semantics "Sports/Diving" will be highlighted in the KG. Thus, for the underwater cameras, the knowledge "Photography" and "Digital Cam" will be less important compared to "Sports Cam". This highlights the importance of identifying and emphasizing relevant rationales in the KG to improve recommendation performance.

In order to achieve accurate and effective knowledge graph-based recommendations, it is important to explicitly model the rationales behind the user preference learning. To address this challenge, we propose a new knowledge graph-enhanced recommender system, called KGRec to leverage attentive knowledge rationalization to generate task-related rational scores for knowledge triplets. KGRec proposes a self-supervised rationale-aware masking mechanism to extract useful rationales from the KG, by adaptively masking knowledge triplets with higher rational scores. By forcing KGRec to learn to reconstruct these important connections, we highlight task-related knowledge rationales. We also align the rational semantics between the KG signals and the Collaborative Filtering (CF) signals via a knowledge-aware contrasting mechanism. This is achieved by filtering out low-scored knowledge that may be potential noise by masking during graph augmentation for contrastive learning. Finally, we inject the rational scores into the knowledge aggregation for the recommendation task, enabling knowledge rational scores to be learned tightly from the CF labels.

In summary, we make the following contributions in this paper:

- We unify generative and contrastive self-supervised learning for knowledge graph-enhanced recommender systems, which enables the distillation of the useful knowledge connections within the knowledge graph for recommendation and align them in a noise-free and rationale-aware manner.
- Our proposed rationale-aware masking mechanism allows us to identify and highlight the most important and relevant information within the knowledge graph, while suppressing potential noise or irrelevant knowledge graph connections.
- To validate the effectiveness of our proposed model, KGRec, we conduct extensive experiments on three real-world datasets. Evaluation results provide strong evidence that our proposed model achieves superior performance compared with existing knowledge-aware recommender systems.

## 2 PRELIMINARIES

We begin by introducing the concepts that will be used in our paper and formally defining the KG-enhanced recommendation task.

**User-Item Interaction Graph.** In a typical recommendation scenario, we have a set of users, denoted by $\mathcal{U}$, and a set of items, denoted by $\mathcal{V}$. Let $u \in \mathcal{U}$ and $v \in \mathcal{V}$ represent a single user and item, respectively. We construct a binary graph $\mathcal{G}_u = (u, y_{uv}, v)$ to denote the collaborative signals between users and items, with $y_{uv} = 1$ if user $u$ interacted with item $v$, and vice versa.

**Knowledge Graph.** We represent real-world knowledge about items with a heterogeneous graph consisting of triplets, denoted by $\mathcal{G}_k = (h, r, t)$. $h, t \in \mathcal{E}$ are knowledge entities, and $r \in \mathcal{R}$ represents the semantic relation connecting them, such as (*author, wrote, book*). It is important to note that the item set is a proper subset of the entity set, *i.e.,* $\mathcal{V} \subset \mathcal{E}$. This allows us to model the complex relationships between items and entities in the KG.

**Task Formulation.** Our KG-aware recommendation task can be formally described as follows: given a user-item interaction graph, denoted by $\mathcal{G}_u$, and a knowledge graph, denoted by $\mathcal{G}_k$, our goal is to learn a recommender model, denoted by $\mathcal{F}(u, v|\mathcal{G}_u, \mathcal{G}_k, \Theta)$, where $\mathcal{F}$ represents the model architecture with learnable parameters $\Theta$. The output of the model is a value in the range $[0, 1]$ that indicates the likelihood of user $u$ interacting with item $v$.

## 3 METHODOLOGY

In this section, we introduce detailed technical design of our proposed KGRec. The overall framework is present in Figure 2.

### 3.1 Rationale Discovery for Knowledge Graph

To automatically distill essential semantics for recommendation from the complex knowledge graph, we propose a rationale weighting function that learns the probability of knowledge triplets being the underlying rationale for collaborative interactions. This rationale function weighs each knowledge triplet based on a learnable graph attention mechanism. Inspired by the heterogeneous graph transformer (HGT) [11], which discriminates the importance of heterogeneous relations, we implement the rationale weighting function $f(h, r, t)$ as follows:

$$f(h, r, t) = \frac{\mathbf{e}_h \mathbf{W}^Q \cdot \left(\mathbf{e}_t \mathbf{W}^K \odot \mathbf{e}_r\right)^\top}{\sqrt{d}}, \tag{1}$$

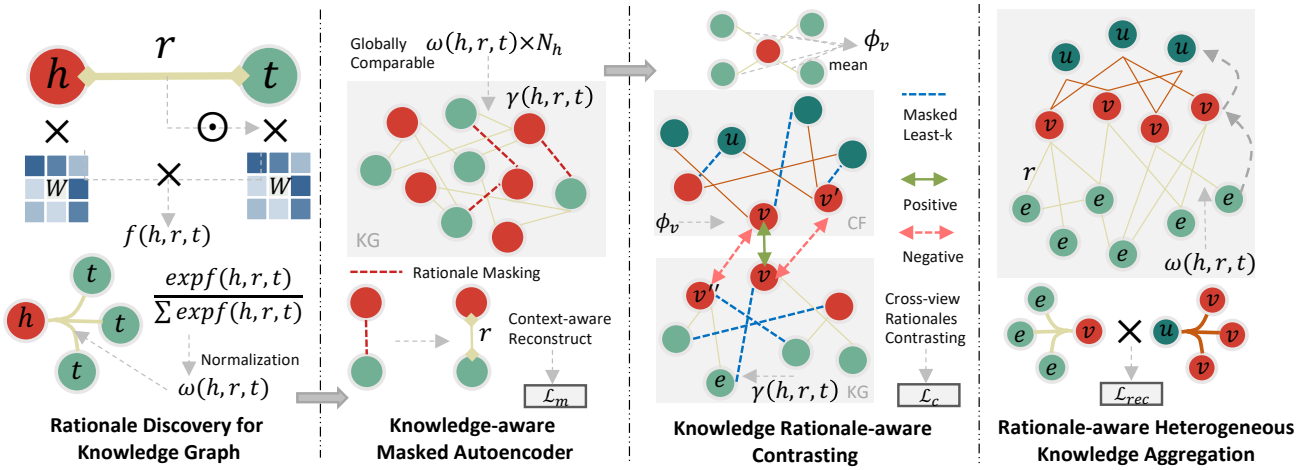



**Figure 2: The overall framework of KGRec. The attentive knowledge rationalization module generates rational scores for KG triplets based on their importance for the recommendation task. Connections with high rational scores are masked, and the model is trained to reconstruct the important connections under relational context. Low-scored KG triplets are considered as noise and removed for rationales contrastive learning between user-item interactions and knowledge graphs.**

Here, $\mathbf{e}_h$, $\mathbf{e}_r$, and $\mathbf{e}_t$ are embeddings for the head, relation, and tail entities, respectively. The trainable weights for attention, $\mathbf{W}^Q$ and $\mathbf{W}^K$, have dimensions of $\mathbb{R}^{d\times d}$, where $d$ is the hidden dimensionality. To model the relational context, we use the element-wise product between the relation $r$ and the tail entity $t$, which corresponds to the rotation of the entity embedding $\mathbf{e}_t$ to the latent space of relation $r$ [25, 35]. The rationale score $f(h,r,t)$ of a knowledge triplet indicates its importance in assisting user preference, as learned by the model and guided by the labels from the recommendation task. To ensure comparability of rationale scores across neighbors of the same head entity, we normalize the scores by the number of neighbors $\mathcal{N}_h$ using the following softmax function:

$$\omega(h,r,t) = \frac{\exp\left(f(h,r,t)\right)}{\sum_{(h,r',t')\in\mathcal{N}_h}\exp\left(f\left(h,r',t'\right)\right)}. \tag{2}$$

## 3.2 Rationale-aware Heterogeneous Knowledge Aggregation

A complex KG often contains a large number of real-world knowledge triplets with heterogeneous nature. Inspired by previous works such as [33, 35, 44], we design an aggregation layer for the knowledge graph that reflects the relational heterogeneity of knowledge triplets. In particular, we focus on the rationales of knowledge triplets, which enable dynamic weighting considering the importance of neighbor entities. To build the knowledge aggregator, we inject the relational context into the embeddings of the neighboring entities, weighting them with the knowledge rationale scores.

$$\mathbf{e}_h^{(l)} = \frac{1}{|\mathcal{N}_h|}\sum_{(h,r,t)\in\mathcal{N}_h}\omega(h,r,t)\mathbf{e}_r\odot\mathbf{e}_t^{(l-1)}, \tag{3}$$

where $l$ denotes the layer of the aggregator, and $\mathcal{N}_h\subseteq\mathcal{G}_k$ is the node-centric sub-graph of first-order neighbors. To inject relational context, we use the same element-wise product as in Equation 1 to bridge the gap between aggregation and rationale weighting. By performing such aggregation across the entire knowledge graph, we carefully consider the contextual relationships between knowledge entities and weight neighbor information for the head entity according to normalized rationale scores.

It's worth noting that items are a subset of knowledge entities. Therefore, we obtain knowledge-aware item representations by aggregating paths $\mathbf{e}_v \leftarrow \mathbf{e}_{t_1} \leftarrow \cdots \leftarrow \mathbf{e}_{t_n}$ on the KG using Equation 3. To model collaborative signals between users and items, we take into account the role of users in the interaction graph $\mathcal{G}_u$. This allows us to generate user embeddings by aggregating the embeddings of the neighboring items in the user-item interaction graph. Specifically, we use a neighbor aggregation method to obtain the user embedding with the following formulas:

$$\mathbf{e}_u^{(l)} = \frac{1}{|\mathcal{N}_u|}\sum_{i\in\mathcal{N}_u}\mathbf{e}_v^{(l-1)}, \tag{4}$$

where $\mathbf{e}_u$ and $\mathbf{e}_v$ represent the user embedding and item embedding, respectively. It's important to note that an item $v$ is equivalent to a certain entity $h, t$ in the knowledge graph.

We can define the final representations of users and entities as the summation of aggregated embeddings from different layers:

$$\mathbf{e}_h = f_k(\mathcal{G}_k;h) = \sum_l^L \mathbf{e}_h^{(l)};\ \mathbf{e}_u = f_u(\mathcal{G}_u;u) = \sum_l^L \mathbf{e}_u^{(l)}, \tag{5}$$

$L$ denotes the number of graph aggregation layers, and $f_*(\cdot,\cdot)$ is the function that generates user or entity representations based on the input graph $\mathcal{G}_u$ or $\mathcal{G}_k$, and certain instances $u$ or $h$.

## 3.3 Knowledge-aware Masked Autoencoder

*3.3.1* **Rationale Masking Mechanism.** As related works have revealed [21, 23, 42], noisy or irrelevant connections between entities in knowledge graphs can lead to suboptimal representation learning. This issue can be particularly problematic in knowledge-aware recommendation systems, where user and item representations are

further interrupted by KG noises [26, 44], resulting in inaccurate recommendation results. To eliminate the noise effect in the KG and distill informative signals that benefit the recommendation task, we propose to highlight important knowledge triplets with high rationale scores, as learned in Equation 1.

Recent studies on masked autoencoders [4, 6, 40] have demonstrated the effectiveness of this approach in enabling models to acquire useful implicit semantics by masking important information during the reconstruction of missing knowledge. Building on these findings, we have designed a generative self-supervised learning task that follows a masking-and-reconstructing approach. During each training step, we mask a batch of knowledge triplets in the KG and reconstruct these relational edges towards a generative self-supervised objective. Additionally, we ensure that the masked triplets have globally high rationale scores, meaning that we mask knowledge that is important for the recommendation task and force the model to learn to reconstruct these connections to highlight useful knowledge for encoding user preference.

To obtain a global measure of the rationale importance of knowledge triplets, we design a criterion. In Equation 2, $\omega(h, r, t)$ reflects the local importance of the triplet among all edges to the same head entity $h$. However, the degree of the head entity can influence the value of $\omega$, making it difficult to compare the importance of triplets across the entire KG. To address this issue, we adjust $\omega(h, r, t)$ by multiplying it with the number of head entity neighbors $|\mathcal{N}_h|$. This modification ensures that the importance of the triplet is weighted by the number of connections of the head entity, rather than just its degree. The updated equation is:

$$\gamma(h, r, t) = |\mathcal{N}_h| \cdot \omega(h, r, t) = \frac{|\mathcal{N}_h| \cdot \exp\left(f(h, r, t)\right)}{\sum_{(h,r',t') \in \mathcal{N}_h} \exp\left(f\left(h, r', t'\right)\right)}. \quad (6)$$

The motivation behind this criterion is to identify the most valuable knowledge triplets across the entire KG. By using the rationale score after softmax, we can determine the relative proportion of a knowledge triplet among its head entity neighbors $\mathcal{N}_h$. We multiply the rationale score with the number of head entity neighbors $|\mathcal{N}_h|$, which makes it globally comparable. By using this approach, we can select the most valuable knowledge triplets across the entire KG based on the value of $\gamma(h, r, t)$. To improve sampling robustness, we add Gumbel noise [13] to the learned rationale scores.

$$\gamma(h, r, t) = \gamma(h, r, t) - \log\left(-\log(\epsilon)\right); \quad \epsilon \sim \text{Uniform}\left(0, 1\right), \quad (7)$$

where $\epsilon$ is a random variable sampled from uniform distribution. Then, we generate a set of masked knowledge triplets by selecting the top $k$-highest rational scores in the KG:

$$\mathcal{M}_k = \{(h, r, t) | \gamma(h, r, t) \in \text{topk}(\Gamma; k_m)\}, \quad (8)$$

where $\Gamma$ represents the distribution of all $\gamma(h, r, t)$. Finally, to create an augmented knowledge graph, denoted by $\mathcal{G}_k^m$, we remove the edges $\mathcal{M}_k$ with low rationale scores from the original knowledge graph $\mathcal{G}_k$. In other words, $\mathcal{G}_k^m$ is obtained by subtracting the set of edges $\mathcal{M}_k$ from the set of edges in $\mathcal{G}_k$, represented by $\mathcal{G}_k \setminus \mathcal{M}_k$.

*3.3.2* **Reconstructing with Relation-aware Objective.** In order to enable our model to recover crucial knowledge in a self-supervised way, we provide the model with entity embeddings created from the augmented graph $\mathcal{G}_k^m$, and train the model to reconnect the masked knowledge edges. Therefore, we begin by applying rationale-aware knowledge aggregation, as outlined in Equation 3, on $\mathcal{G}_k^m$ to produce entity embeddings, in which $k_m$ rationale edges have been removed.

$$\mathbf{e}_h = f_k(\mathcal{G}_k^m; h); \ \mathbf{e}_t = f_k(\mathcal{G}_k^m; t), \quad (9)$$

The function $f_k(\cdot)$ is the aggregation function on the knowledge graph, as defined in Equation 5. At this point, the knowledge triplets with significant rationale scores, denoted by $\mathcal{M}_k$, which were not visible during the aggregation stage, can be used as self-supervision labels for reconstruction. Given the rich relational heterogeneity in the knowledge graph, we aim to reconstruct the important rational connections under relational contexts. To achieve this, we minimize the following dot-product log-loss for the label triplets, with $\sigma(\cdot)$ representing the sigmoid activation function:

$$\mathcal{L}_m = \sum_{(h,r,t) \in \mathcal{M}_k} -\log\left(\sigma\left(\mathbf{e}_h^\top \cdot (\mathbf{e}_t \odot \mathbf{e}_r)\right)\right). \quad (10)$$

## 3.4 Knowledge Rationale-aware Contrasting

*3.4.1* **Rationale-aware Graph Augmentation.** As explained earlier, the hierarchical rationales for knowledge triplets are derived from the connection between the knowledge graph and user-involved recommendation labels. In order to further enhance the interpretability of the knowledge rationalization modules, we draw inspiration from previous works [52]. Specifically, we propose to align the representations of the knowledge graph with collaborative filtering signals, which allows us to explicitly model cross-view rationales. To construct debiased contrastive views, we begin by identifying and removing weakly task-related edges that could potentially introduce noise in both graphs.

Regarding the knowledge graph, it is worth noting that knowledge triplets with lower rationale scores tend to have less impact on the recommendation task. Consequently, we aim to improve the quality of the graph by removing the noisy triplets. This augmentation process ensures that the remaining triplets are more informative and have a higher rationale score. By doing so, we can enhance the performance of our model and better capture the underlying relationships between the entities in the graph.

$$\mathcal{S}_k = \{(h, r, t) | \gamma(h, r, t) \in \text{topk}(-\Gamma; \rho_k)\}; \ \mathcal{G}_k^c = \mathcal{G}_k \setminus \mathcal{S}_k, \quad (11)$$

In Equation 8, we introduced the knowledge attentive scores $\gamma$ and $\Gamma$, which are computed with the addition of Gumbel noise. Here, $\Gamma$ represents the distribution of all $\gamma$ values. By taking the negative of $\Gamma$, denoted as $-\Gamma$, we can use the top-k function to calculate the least-k values. The hyperparameter $\rho_k$ controls the dropout ratio during training. We also introduce the augmented knowledge graph $\mathcal{G}_k^c$, which is debiased from noise with lower rationale scores.

In addition to the knowledge graph, we also aim to improve the quality of the u-i interaction graph by removing noisy interactions that are not conducive to cross-view alignment. Specifically, we want to retain interaction edges that clearly reflect the user's interests and can better guide knowledge graph rationalization through cross-view contrasting. Given that the semantics of item embeddings can be influenced by their linked knowledge in the KG, we propose to weight each interaction edge by considering the rationales of the knowledge triplets connected to the item. This

approach allows us to better reflect the noise associated with each interaction edge. To implement this, we calculate the mean value of the rationale scores for all the knowledge triplets linked to the item. This mean value is then used as a weight for the corresponding interaction edge, which helps to distinguish between informative and noisy interactions.

$$\phi_v = \text{mean}(\{\gamma(h,r,t)|h = v \vee t = v\}). \tag{12}$$

A lower $\phi_v$ value implies that the knowledge entities neighboring an item in the KG are relatively less contributive to the recommendation task, which can lead to bias in the item representation. To address this issue, we filter our interaction edges using the $\phi_v$ score and augment the graph with only the informative interactions. To avoid overfitting on user and item representations, we adopt a multinomial distribution sampling strategy [12, 17] to derive more randomized samples for edge dropout. This approach helps to ensure that the model is not overly reliant on a specific set of interactions and can generalize well to new data. Formally, the process can be defined as follows:

$$\phi'_v = \frac{\exp \phi_v}{\sum_v \exp \phi_v};\ \mathcal{S}_u \sim \text{multinomialNR}(\Phi'; \rho_u), \tag{13}$$

After calculating the $\phi_v$ score for each item $v$, which represents the mean value of the rationale scores for all the knowledge triplets linked to the item, we apply softmax to obtain a probability distribution $\phi'$ over all items. The resulting distribution $\Phi'$ is used to sample a subset of items without replacement using the multinomial distribution sampling method, denoted as multinomialNR$(\cdot;\cdot)$. Here, $\rho_u$ denotes the size of the sampled candidates. By following the previous definitions, we can generate the augmented u-i graph as the difference between the original u-i graph $\mathcal{G}_u$ and the set of sampled interactions $\mathcal{S}_u$, *i.e.,* $\mathcal{G}^c_u = \mathcal{G}_u \setminus \mathcal{S}_u$.

*3.4.2* **Contrastive Learning with Cross-View Rationales.** With the augmented knowledge graph and u-i graph, we use pre-defined aggregators to capture the view-specific node representations for items as the contrastive embeddings. For the u-i interaction view, we utilize the state-of-the-art LightGCN [8] module to iteratively capture high-order information on $\mathcal{G}^c_u$.

$$\mathbf{x}^{(l)}_u = \sum_{v \in \mathcal{N}_u} \frac{\mathbf{x}^{(l-1)}_v}{\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}};\ \mathbf{x}^{(l)}_v = \sum_{u \in \mathcal{N}_v} \frac{\mathbf{x}^{(l-1)}_u}{\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}}. \tag{14}$$

We obtain the final contrastive embeddings for items in the u-i view by summing up the representations from all layers of the LightGCN module. For the augmented knowledge graph, we use a rationale-aware knowledge aggregation mechanism to generate the knowledge-view item representations, which take into account the rationale scores associated with the knowledge triplets.

$$\mathbf{x}^k_v = f_k(\mathcal{G}^c_k; v). \tag{15}$$

It is important to note that the contrastive embeddings $\mathbf{x}^u_i$ and $\mathbf{x}^k_i$ are from different representation spaces, namely the collaborative relational signals and knowledge graph signals. We feed them into two different MLPs to map them into the same latent space.

$$\mathbf{z}^*_v = \sigma\left(\mathbf{x}^{*\mathsf{T}}_v \mathbf{W}^*_1 + \mathbf{b}^*_1\right)^\mathsf{T} \mathbf{W}^*_2 + \mathbf{b}^*_2, \tag{16}$$

where the notation $* \in u, k$ denotes view-specific representations, namely $\mathbf{z}^u_v$ and $\mathbf{z}^k_v$. The learnable weights and bias denoted as $\mathbf{W}$ and $\mathbf{b}$. By doing so, we can effectively capture the complementary information from both views.

To ensure the alignment of cross-view item representations, we adopt a contrastive objective. To avoid over-fitting and eliminate the false-negative effect, as inspired by [31], we modify the widely used InfoNCE [5] loss by specifying one random sample for each view as the negative. Formally, we define our contrastive loss as:

$$\mathcal{L}_c = \sum_{v \in \mathcal{V}} -\log \frac{\exp(s(\mathbf{z}^u_v, \mathbf{z}^k_v)/\tau)}{\sum_{j \in \{v, v', v''\}} (\exp(s(\mathbf{z}^u_v, \mathbf{z}^k_v)/\tau) + \exp(s(\mathbf{z}^u_j, \mathbf{z}^k_v)/\tau))}, \tag{17}$$

In the contrastive loss, $v'$ and $v''$ are stochastically sampled negative candidates for item $v$. The similarity measurement $s(\cdot)$ is set to the cosine similarity of normalized vectors. The temperature hyper-parameter $\tau$ controls the hardness of the contrastive goal [14, 39].

## 3.5 Model Learning and Discussion

For the main recommendation task, we use the dot product between the user and item representations as the prediction, which is denoted as $\hat{y}_{uv} = \mathbf{e}^\mathsf{T}_u \mathbf{e}_v$. To optimize the model parameters, we adopt the widely used Bayesian Personalized Ranking (BPR) loss to optimize the model parameters as follows:

$$\mathcal{L}_{rec} = \sum_{(u,v,j) \in \mathcal{D}} -\log \sigma\left(\hat{y}_{uv} - \hat{y}_{uj}\right), \tag{18}$$

In the BPR loss, we use the training instances $\mathcal{D} = (u, v, j)$, where $v$ is the ground-truth and $j$ is a randomly sampled negative interaction. It is worth noting that we continue to use the entity embeddings $\mathbf{e}_v$ from the masked graph $\mathcal{G}^m_k$ for the recommendation task, rather than performing aggregation on the original knowledge graph again. This is because the masked triplets are generally of small size (*e.g.,* 512) compared to the whole graph (*e.g.,* millions), and this trick can greatly improve the training efficiency while affecting the representation learning only minimally. Moreover, according to [6, 18], this setting can increase the difficulty of the main task learning and improve the optimization effect.

To optimize all three loss functions, we use a joint learning approach with the following overall loss function:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_1 \mathcal{L}_m + \lambda_2 \mathcal{L}_c, \tag{19}$$

where $\lambda_1$ and $\lambda_2$ represent the weight of the mask-and-reconstruction and cross-view contrastive learning tasks, respectively. We omit the notation of L2 regularization terms for brevity.

*3.5.1* **Connection to Alignment and Uniformity.** Investigating the alignment and uniformity merits of learned representations by the generative and contrastive tasks is important for providing fundamental support for the proposed KGRec method. Following [32, 47], the mathematical definitions for the uniformity and alignment of learned vector representations are presented below:

$$\mathcal{L}_{align} \coloneqq \mathbb{E}_{(x,y)\sim p^+}[\|\mathbf{x} - \mathbf{y}\|^\alpha_2];\ \alpha > 0 \tag{20}$$

$$\mathcal{L}_{uni} \coloneqq \log \mathbb{E}_{(x,y) \overset{i.i.d.}{\sim} p}[\exp(-\gamma\|\mathbf{x} - \mathbf{y}\|^2_2)];\ \gamma = 1/2\sigma^2, \tag{21}$$

where the exponent $\alpha$ of the Euclidean distance controls the degree to which the learning algorithm focuses on aligning the learned

representations with the positive labels. The distribution $p$ of training data and $p^+$ distribution of positive labels are used to compute the expected value of the alignment loss.

We first prove that the rational masking-reconstructing task is an explicit alignment for features. According to the generative loss in Equation 10, the optimization of $\mathcal{L}_m$ equals to:

$$\min \mathbb{E}_{(x,y)\sim p_r^+} \left[ \sum_{x,y} \log \left( \sigma \left( \mathbf{x}^\mathsf{T} \mathbf{y} \right) \right) \right], \tag{22}$$

where the variable $\mathbf{x}$ corresponds to the feature vector of the head entity $\mathbf{e}_h$, while the variable $\mathbf{y}$ corresponds to the element-wise product of the feature vectors of the tail entity and relation, given by $\mathbf{e}_r \odot \mathbf{e}_t$. The distribution $p_r^+$ represents the set of knowledge rationales with high rational scores. Note that:

$$\|\mathbf{x} - \mathbf{y}\|_2^\alpha = (2 - 2 \cdot \mathbf{x}^\mathsf{T} \mathbf{y})^{\frac{\alpha}{2}}. \tag{23}$$

Since the generative loss function $\mathcal{L}_m$ is in the form of an alignment loss, as defined in Equation 20, minimizing it leads to the alignment of the masked rationale knowledge triplets.

We can further show that the contrastive loss in Equation 17 reflects the alignment and uniformity properties. Considering that:

$$\mathcal{L}_c = \mathbb{E}_{\substack{(x,y)\sim p_c^+ \\ \{x_i^-\}_{i=1}^2 \sim p_c}} \left[ -\frac{1}{\tau} \mathbf{x}^\mathsf{T} \mathbf{y} + \log \left( \exp(\mathbf{x}^\mathsf{T} \mathbf{y}/\tau) + \sum_i \exp(\mathbf{x_i}^{-\mathsf{T}} \mathbf{x}/\tau) \right) \right] \tag{24}$$

$$\geq \mathbb{E}_{\substack{x\sim p_c \\ \{x_i^-\}_{i=1}^2 \sim p_c}} \left[ -\frac{1}{\tau} + \log \left( \exp(\frac{1}{\tau}) + \sum_i \exp(\mathbf{x_i}^{-\mathsf{T}} \mathbf{x}/\tau) \right) \right] \tag{25}$$

The positive pair in Equation 17 is denoted as $\mathbf{x}, \mathbf{y}$, and the negative samples are denoted as $\mathbf{x}^-$ for brevity. The set of random negative samples $x_i^-{}_{i=1}^2 \sim p_c$ in Equation 17 is drawn from the distribution $p_c$ of cross-view item representations. As a result, the lower bound of the contrastive loss function $\mathcal{L}_c$ in Equation 25 is satisfied only if the embeddings $\mathbf{x}, \mathbf{y}$ are perfectly aligned, i.e., $\mathbf{x}^\mathsf{T} \mathbf{y} = 1$, which is equivalent to the definition of alignment in Equation 20. If the embeddings satisfy the perfect alignment condition, the optimization of $\mathcal{L}_c$ simplifies to a degenerate form.

$$\min \mathbb{E}_{\substack{x\sim p_c \\ \{x_i^-\}_{i=1}^2 \sim p_c}} \left[ \log \left( \sum_i \exp(\mathbf{x_i}^{-\mathsf{T}} \mathbf{x}/\tau) \right) \right], \tag{26}$$

The alignment and uniformity properties in the generative loss function $\mathcal{L}_m$ and the contrastive loss function $\mathcal{L}_c$ can benefit representation learning by ensuring that positive pairs are in agreement and that random instances are pushed as negatives. In addition, the proposed knowledge rationalization improves the sampling distribution to be rational and noise-resistant, instead of using a random distribution as in the original forms. By exploiting rationales in the KG, we empower the alignment property with rationality-aware positive pairing ability, which provides better gradients for model learning. Additionally, for cross-view rationales, we remove potential noise to build a noise-free distribution, which eliminates the effect of false negative pairing and improves the contrastive effectiveness. Overall, our KGRec is able to derive better alignment and uniformity compared to stochastic methods, which can lead to improved representation for more accurate recommendations.

**Table 1: Statistics of Three Evaluation Datasets.**

| Statistics | Last-FM | MIND | Alibaba-iFashion |
|---|---|---|---|
| # Users | 23,566 | 100,000 | 114,737 |
| # Items | 48,123 | 30,577 | 30,040 |
| # Interactions | 3,034,796 | 2,975,319 | 1,781,093 |
| # Density | 2.7e-3 | 9.7e-4 | 5.2e-4 |
| Knowledge Graph | | | |
| # Entities | 58,266 | 24,733 | 59,156 |
| # Relations | 9 | 512 | 51 |
| # Triplets | 464,567 | 148,568 | 279,155 |

# 4 EVALUATION

In this section, we conduct experiments to answer several research questions related to the proposed KGRec framework.

- **RQ1**: Can KGRec outperform state-of-the-art baseline models of different types in terms of recommendation performance?
- **RQ2**: How do the key designs in KGRec contribute to its overall performance, and what is its sensitivity to hyperparameters?
- **RQ3**: What benefits does KGRec bring to tackling task-specific challenges such as cold-start and long-tail item recommendation?
- **RQ4**: Can KGRec derive interpretability with rational scores?

## 4.1 Experimental Setup

*4.1.1* **Dataset.** To ensure a diverse and representative evaluation, we use three distinct datasets that reflect real-life scenarios: Last-FM for music recommendations, MIND for news recommendations, and Alibaba-iFashion for shopping recommendations. We preprocess the datasets using the commonly adopted 10-Core approach to filter out users and items with less than 10 occurrences. To construct the knowledge graphs, we employ different methods for each dataset. For Last-FM, we map the items to Freebase entities and extract knowledge triplets, following the techniques used in [33] and [50]. For MIND, we collect the knowledge graph from Wikidata[1] using the representative entities in the original data, following the approach proposed in [26]. For Alibaba-iFashion, we manually construct the knowledge graph using category information as knowledge, as done in [35]. Table 1 summarizes the statistics of user-item interactions and knowledge graphs for three evaluation datasets.

*4.1.2* **Evaluation Protocols.** To ensure fair evaluation, we employ the full-rank setting and divide our dataset into three parts: 70% for training, 10% for hyperparameter tuning, and 20% for testing. We measure the performance of our proposed KGRec using the Recall@N and NDCG@N metrics, with N set to 20 for top-N recommendations. We implement KGRec using PyTorch and compare its performance with various baseline models using official or third-party code. To optimize the performance of KGRec, we conduct a hyperparameter search for the masking size, keeping proportion for contrastive learning, and temperature value. Specifically, we explore values of masking size from the range of $\{128, 256, 512, 1024\}$, keeping proportion $\rho_k$ and $\rho_u$ from $\{0.4, 0.5, 0.6, 0.7, 0.8\}$, and temperature value from the range of $\{0.1, \cdots, 1.0\}$. The number of GNN layers is set to 2 for all graph-based methods.

[1] https://query.wikidata.org/

*4.1.3* **Baseline Models.** To verify the effectiveness of our proposed design, we conduct benchmark evaluations between KGRec and various baseline models from different research lines.

**General Collaborative Filtering Methods.**

- **BPR** [22] is a matrix factorization method that uses pairwise ranking loss based on implicit feedback.
- **NeuMF** [9] incorporates MLP into matrix factorization to learn the enriched user and item feature interactions.
- **GC-MC** [1] considers recommendation as a link prediction problem on the user-item graph and proposes a graph auto-encoder framework for matrix completion.
- **LightGCN** [8] is a state-of-the-art recommendation method based on graph neural networks (GNNs), which improves performance by removing activation and feature transformation. **SGL** [39] introduces a self-supervised learning paradigm to GNN-based recommendation by using stochastic augmentation on the user-item graph based on the InfoNCE objective.

**Embedding-based Knowledge-aware Recommenders.**

- **CKE** [49] is an embedding-based KG recommender that leverages TransR [20] to enrich item representations by training on structural knowledge, thereby enhancing collaborative filtering.
- **KTUP** [2] trains TransH [37] using preference-injected CF and enables mutual complementation between CF and KG signals.

**GNN-based Knowledge Graph-enhanced Recommenders.**

- **KGNN-LS** [28] considers user preferences towards different knowledge triplets in graph convolution and introduces label smoothing as regularization to force similar user preference weights between nearby items in the KG.
- **KGCN** [30] aggregates knowledge for item representations by considering high-order information with GNN and uses preferences from user embeddings as weights.
- **KGAT** [33] introduces the concept of a collaborative KG to apply attentive aggregation on the joint user-item-entity graph, with attention scores reflecting the importance of knowledge triplets.
- **KGIN** [35] is a state-of-the-art method that models user intents for relations and employs relational path-aware aggregation to effectively capture rich information on the knowledge graph.

**Self-Supervised Knowledge-aware Recommenders.**

- **MCCLK** [52] performs contrastive learning in a hierarchical manner for data augmentation, so as to consider structural information for the user-item-entity graph.
- **KGCL** [44] introduces graph contrastive learning for KGs to reduce potential knowledge noise. KG contrastive signals are further used to guide the user preference learning.

## 4.2 RQ1: Overall Performance Comparison

We report the performance of all the methods on three datasets in Table 2. Based on the results, we make the following observations:

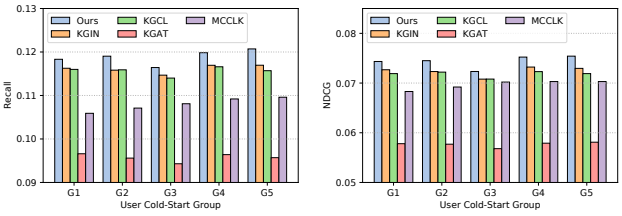
- The proposed KGRec consistently outperforms all baseline models on both metrics and all three datasets. This can be attributed to three factors. First, by using rational masking and reconstruction, KGRec is able to capture knowledge information that is truly useful for the recommendation task. Second, KGRec is equipped with rational cross-view contrastive learning on augmented, noise-free graphs, which allows for better exploitation of the latent relatedness between KG and CF signals. Third, the knowledge aggregation layer is weighted by knowledge rational scores to reflect the different importance of knowledge triplets. Additionally, the superior results on datasets with vastly different statistics suggest that the proposed knowledge rationalization mechanism can automatically discover useful knowledge related to downstream tasks, regardless of the data characteristics.
- On the three datasets, there is no consistent winner among the baseline models. Contrastive learning-based methods (*e.g.,* MCCLK and KGCL) are not always better than non-self-supervised methods (*e.g.,* KGIN). This may be due to the limitations of random graph augmentation or intuitive handcrafted cross-view pairing, which may fail to discover truly useful KG information from the contrastive views for encoding the interests of users.
- GNN-based knowledge-aware recommenders can consistently outperform embedding-based models. This advantage is due to GNNs' ability to capture more complex and higher-order information on the KG, compared to the linear transition-based modeling adopted by embedding-based models.
- The introduction of knowledge graphs does not always lead to better performance in recommendation systems. For instance, methods such as CKE and KTUP typically perform worse than non-KG methods like LightGCN and SGL. Even KGNN-LS and KGCN cannot consistently outperform SGL in some metrics. This effect is more noticeable when the dataset has a complex KG and sparse interactions. We suggest that some KG-aware recommenders struggle to effectively model complex relational paths and mitigate noise in the KG, resulting in suboptimal KG representation learning and worse performances. On the other hand, LightGCN and SGL focus more on resolving the sparsity problem of user-item interactions with self-supervision signals.

**Table 2: The overall performance evaluation results for KGRec and compared baseline models on three experimented datasets, where the best and second-best performances are denoted in bold and borderline, respectively.**

| Model | Last-FM | | MIND | | Alibaba-iFashion | |
|---|---|---|---|---|---|---|
| | Recall | NDCG | Recall | NDCG | Recall | NDCG |
| BPR | 0.0690 | 0.0585 | 0.0384 | 0.0253 | 0.0822 | 0.0501 |
| NeuMF | 0.0699 | 0.0615 | 0.0308 | 0.0237 | 0.0506 | 0.0276 |
| GC-MC | 0.0709 | 0.0631 | 0.0386 | 0.0261 | 0.0845 | 0.0502 |
| LightGCN | 0.0738 | 0.0647 | 0.0419 | 0.0253 | 0.1058 | 0.0652 |
| SGL | 0.0879 | 0.0775 | 0.0429 | 0.0275 | 0.1141 | 0.0713 |
| CKE | 0.0845 | 0.0718 | 0.0387 | 0.0247 | 0.0835 | 0.0512 |
| KTUP | 0.0865 | 0.0671 | 0.0362 | 0.0302 | 0.0976 | 0.0634 |
| KGNN-LS | 0.0881 | 0.0690 | 0.0395 | 0.0302 | 0.0983 | 0.0633 |
| KGCN | 0.0879 | 0.0694 | 0.0396 | 0.0302 | 0.0983 | 0.0633 |
| KGAT | 0.0870 | 0.0743 | 0.0340 | 0.0287 | 0.0957 | 0.0577 |
| KGIN | 0.0900 | 0.0779 | 0.0357 | 0.0225 | 0.1144 | 0.0723 |
| MCCLK | 0.0671 | 0.0603 | 0.0327 | 0.0194 | 0.1089 | 0.0707 |
| KGCL | 0.0905 | 0.0769 | 0.0399 | 0.0247 | 0.1146 | 0.0719 |
| KGRec | **0.0943** | **0.0810** | **0.0439** | **0.0319** | **0.1188** | **0.0743** |

**Table 3: Ablation results of KGRec with different variants. The superscript * denotes the largest change in performance.**

| Ablation Settings | Last-FM | | MIND | | Alibaba-iFashion | |
|---|---|---|---|---|---|---|
| | Recall | NDCG | Recall | NDCG | Recall | NDCG |
| KGRec | **0.0943** | **0.0810** | **0.0439** | **0.0319** | **0.1188** | **0.0743** |
| w/o MAE | 0.0918* | 0.0792* | 0.0374* | 0.0238* | 0.1178* | 0.0737* |
| w/o Rationale-M | 0.0929 | 0.0805 | 0.0423 | 0.0311 | 0.1183 | 0.0739 |
| w/o CL | 0.0926 | 0.0796 | 0.0425 | 0.0313 | 0.1180 | 0.0734 |
| w/o Rationale-Aug | 0.0931 | 0.0801 | 0.0405 | 0.0278 | 0.1185 | 0.0741 |




(a) Alibaba-iFashion@Recall (b) Alibaba-iFashion@NDCG

**Figure 3: Evaluation results on different user groups. Lower group number implies stronger cold-start effect.**

## 4.3 RQ2: Ablation Study

*4.3.1* **Key Module Ablation.** In this study, we investigate the effectiveness of key modules in our proposed KGRec from the perspectives of our designed rational masked autoencoding and contrastive learning for recommendation. To compare with the original method, we built four model variants, including:

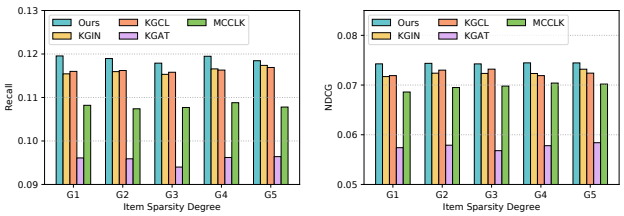
- w/o MAE: removing the generative SSL task of rationale-aware knowledge graph masking and reconstruction.
- w/o Rationale-M: replacing the rationale knowledge masking with random masking while keeping the masking size unchanged.
- w/o CL: disabling the cross-view contrastive learning task.
- w/o Rationale-Aug: replacing the rational graph augmentation with random masking while keeping the masking size unchanged.

We report the results of the ablation study in Table 3 and make the following observations: i) The proposed rationale knowledge masking and reconstruction contributes the most to performance enhancement. This demonstrates that mask&reconstruction is an effective strategy for exploiting highly useful knowledge triplets for recommendation. ii) The rational masking mechanism for both reconstruction and contrastive learning can further improve performance by selecting valuable information and dropping informative knowledge. iii) The contrastive learning is also beneficial for performance. However, we observed that adding non-rationale augmented graph contrastive learning on the MIND dataset can hurt performance. This indicates that simple intuitive cross-view contrasting is not always effective due to noises in the graph.

*4.3.2* **Sensitivity to Key Hyperparameters.** We present our results and discussion of parameter study in Appendix A.1.

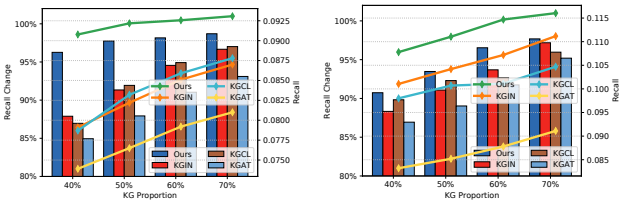
## 4.4 RQ3: Model Benefits Investigation

*4.4.1* **Cold-start Recommendation.** We conduct a study to evaluate the effectiveness of KGRec in addressing the common cold-start problem in recommendation systems. We divided users in the Alibaba-iFashion dataset into five groups based on the number of interactions, with smaller group numbers indicating stronger cold-start effects. We then separately tested the performance of KGRec and several strong baselines in each group and reported the results in Figure 3. Our findings demonstrate that KGRec outperforms other baseline methods in all cold-start groups, indicating its effectiveness in addressing the cold-start problem for a diverse range of users. This can be attributed to the design of the rationale knowledge masked autoencoding and rationale-based cross-view contrastive learning, which highlight useful knowledge for representation learning and contrast cross-view signals. Therefore, KGRec can effectively alleviate cold-start issue.




(a) Alibaba-iFashion@Recall (b) Alibaba-iFashion@NDCG

**Figure 4: Evaluation results on different item sparsity levels.**




(a) Last-FM (b) Alibaba-iFashion

**Figure 5: Evaluation results on different KG proportions.**

*4.4.2* **Long-tail Item Recommendation.** We investigate whether KGRec can improve representation learning for long-tail items. We counted the occurrence of each item and divided all users into five groups based on the average sparsity degree of items they interacted with. The results are reported in Figure 4. Our findings demonstrate that KGRec consistently outperforms baseline models across different groups, indicating its effectiveness in addressing data scarcity problems. This can be attributed to the design of rationale mining, which allows KGRec to better leverage external knowledge and improve representation learning for long-tail items.

*4.4.3* **Recommendation with small proportion of KG.** We evaluate KGRec's capacity in highlighting important task-related connections from the knowledge graph. Specifically, we tested the recommendation performance of KGRec and baseline models under partial knowledge graphs with different keeping ratios ranging from 40% to 70%. We randomly selected a proportion of knowledge triplets from the original KG in the Last-FM and Alibaba-iFashion datasets for knowledge aggregation, and the results are reported in Figure 5. Our findings demonstrate that KGRec can still maintain considerable performance (>95% on Last-FM and >90% on Alibaba-iFashion) with only a small portion of KG. Compared to baseline models, KGRec shows minimal performance degradation in all cases. This can be attributed to the design of rationale knowledge masking

**Table 4: KG Relations with highest average global rationale scores for news categories learned on MIND dataset.**

| Category | Relation (Wiki ID) | Avg. Score |
|---|---|---|
| *sports* | member of sports team (P54) | 1.235 |
| | league of (P118) | 1.117 |
| *newspolitics* | member of political party (P102) | 1.341 |
| | position held (P39) | 1.097 |
| *travel* | part of (P361) | 1.105 |
| | located in (P131) | 1.190 |
| *finance* | owned of (P1830) | 1.203 |
| | stock exchange (P414) | 1.157 |
| *tv-celebrity* | award received (P166) | 1.084 |
| | cast member (P161) | 1.139 |

and reconstruction mechanism, which can effectively distill useful knowledge from the given portion of the KG. Additionally, the cross-view contrastive learning can enhance KG learning with CF signals to alleviate the absence of some knowledge. Overall, the results further validate the rationality of KGRec's design.

### 4.5 RQ4: Model Explainability Study

We discuss the explainability of results generated by KGRec in Appendix A.2, which provides insights into how KGRec incorporates the KG and rationale knowledge for enhancing recommendation.

## 5 RELATED WORK

### 5.1 Knowledge-aware Recommender Systems

Knowledge graphs are valuable sources of side information for item representation learning and user modeling in recommender systems. Currently, knowledge-aware recommendation methods can be generally categorized into three groups: embedding-based methods, path-based methods, and GNN-based methods. i) Embedding-based methods [2, 27, 29, 49] incorporate knowledge graph entity embedding into user and item representations to enhance the recommendation learning. For example, CKE [49] proposes to integrate the modeling of different types of side information for items with collaborative filtering. It encodes a knowledge graph with the transitive KG completion method TransR [20] as part of item representations. ii) Path-based methods [10, 36, 48] focus on exploiting the rich semantics in relational meta-paths on the KG. For instance, KPRN [36] adopts an LSTM to model the extracted meta-paths and aggregates user preference along each path by fully-connected layers. iii) GNN-based methods [30, 33] extend GNNs to model the KG and use the learned representations for recommendation. For example, KGAT [33] proposes to use a graph attention mechanism to propagate user and item embeddings on the KG, and then apply a multi-layer perceptron to produce the final recommendation score.

The line of GNN-based knowledge-aware recommenders [28, 30, 33, 35] aims to unify the two paradigms and combine their strengths. GNNs have a powerful ability to capture high-order information, making them effective at extracting useful information from the KG. KGCN [30] samples a fixed number of neighbors as the receptive field to aggregate item representations on the KG. KGAT [33] leverages graph attention networks (GATs) to weight the knowledge aggregation on the KG by considering the different importance of knowledge neighbors. KGIN [35] further considers user latents towards different relations in the KG and injects relational embedding in the aggregation layer to improve performance. GNN-based methods are currently the state-of-the-art solutions due to their ability to exploit rich semantics from the graph and their considerable efficiency.

### 5.2 Self-Supervised Recommendation

Incorporating self-supervised learning (SSL) techniques into recommender systems has become a new trend in the research community to address inherent data sparsity problems by leveraging additional supervision signals from raw data [16, 51]. Existing studies have explored various SSL techniques for different recommendation tasks. For large-scale industry applications, [45] introduces contrastive learning in the two-tower architecture for feature augmentation with the proposed correlated feature masking strategy. SGL [39] applies graph contrastive learning to graph collaborative filtering using random augmentation on graphs such as node dropout, edge dropout, and random walk to generate contrastive views and enforce agreement with InfoNCE loss. For sequential recommendation, S3Rec [51] aims to augment the sequence itself by masking and adopts the contrast between augmented sequences as an auxiliary task. For social recommendation, MHCN [46] performs contrastive learning between user embedding and its social embedding extracted from a sub-hypergraph of the social network. For multi-modal recommender systems, MMSSL [38] aims to provide a universal solution for capturing both modality-specific collaborative effects and cross-modality interaction dependencies, allowing for more accurate recommendations.

KGCL [44] develops graph contrastive learning on the KG to alleviate noise and long-tail problems, while also leveraging additional signals from KG agreement to guide user/item representation learning. MCCLK [52] employ cross-view contrastive learning between the KG and interaction graph to mitigate sparse supervision signals. However, we argue that these methods do not sufficiently consider the rationales embedded in the KG. By explicitly rationalizing knowledge triplets for recommendation, our KGRec achieves a significant performance improvement compared to these methods.

## 6 CONCLUSION

In this paper, we presented a novel graph self-supervised rationalization method (KGRec) for knowledge-aware recommendation. Our motivation is rooted in the hierarchical rationality of knowledge triplets. We build our method on the attentive knowledge rationalization to weight knowledge triplets, and introduce a novel rational masking and reconstruction module to emphasize rational knowledge. The rational scores were further used to facilitate the knowledge-aware cross-view contrastive learning, where low-scored less informative knowledge was filtered out as noise. Results of extensive experiments validate the advantages of KGRec against state-of-the-art solutions. In future works, we will explore more complex methods for knowledge graph rationalization, such as graph structure learning and graph sparsification. This direction can potentially provide more insights into the underlying knowledge graph structure.

# REFERENCES


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

[2] 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 *The Web Conference (WWW)*. 151–161.

[3] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 335–344.

[4] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. 2022. Masked Autoencoders As Spatiotemporal Learners. *arXiv preprint arXiv:2205.09113* (2022).

[5] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 297–304.

[6] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16000–16009.

[7] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 355–364.

[8] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 639–648.

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

[10] Binbin Hu, Chuan Shi, et al. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. 1531–1540.

[11] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *The Web Conference (WWW)*. 2704–2710.

[12] Tiansheng Huang, Weiwei Lin, Li Shen, Keqin Li, and Albert Y Zomaya. 2022. Stochastic client selection for federated learning with volatile clients. *IEEE Internet of Things Journal* (2022).

[13] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparametrization with Gumble-Softmax. In *International Conference on Learning Representations (ICLR)*.

[14] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. (2020), 18661–18673.

[15] Daniel Kluver, Michael D Ekstrand, and Joseph A Konstan. 2018. Rating-based collaborative filtering: algorithms and evaluation. *Social information access: Systems and technologies* (2018), 344–390.

[16] Chaoliu Li, Lianghao Xia, Xubin Ren, Yaowen Ye, Yong Xu, and Chao Huang. 2023. Graph Transformer for Recommendation. *arXiv preprint arXiv:2306.02330* (2023).

[17] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* 2 (2020), 429–450.

[18] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. 2022. Scaling Language-Image Pre-training via Masking. *arXiv preprint arXiv:2212.00794* (2022).

[19] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, et al. 2018. Variational autoencoders for collaborative filtering. In *The Web Conference (WWW)*. 689–698.

[20] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[21] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *AAAI Conference on Artificial Intelligence (AAAI)*. 2901–2908.

[22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*. 452–461.

[23] Baoxu Shi and Tim Weninger. 2018. Open-world knowledge graph completion. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[24] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *International Conference on Information and Knowledge Management (CIKM)*. 1441–1450.

[25] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. [n. d.]. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations (ICLR)*.

[26] Yu Tian, Yuhao Yang, Xudong Ren, Pengfei Wang, Fangzhao Wu, Qian Wang, and Chenliang Li. 2021. Joint knowledge pruning and recurrent graph convolution for news recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 51–60.

[27] Hongwei Wang, Fuzheng Zhang, et al. 2018. DKN: Deep knowledge-aware network for news recommendation. In *The Web Conference (WWW)*. 1835–1844.

[28] 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 *International Conference on Knowledge Discovery & Data Mining (KDD)*. 968–977.

[29] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *ACM Web Conference (WWW)*. 2000–2010.

[30] Hongwei Wang, Miao Zhao, Xing Xie, et al. 2019. Knowledge graph convolutional networks for recommender systems. In *The Web Conference (WWW)*. 3307–3313.

[31] Kai Wang, Yu Liu, and Quan Z Sheng. 2022. Swift and Sure: Hardness-aware Contrastive Learning for Low-dimensional Knowledge Graph Embeddings. In *The Web Conference (WWW)*. 838–849.

[32] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning (ICML)*. 9929–9939.

[33] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. 950–958.

[34] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 165–174.

[35] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, et al. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *The Web Conference (WWW)*. 878–887.

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

[37] Zhen Wang, Jianwen Zhang, et al. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[38] Wei Wei, Chao Huang, Lianghao Xia, and Chuxu Zhang. 2023. Multi-Modal Self-Supervised Learning for Recommendation. In *The Web Conference (WWW)*. 790–800.

[39] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 726–735.

[40] Lianghao Xia, Chao Huang, Chunzhen Huang, Kangyi Lin, Tao Yu, and Ben Kao. 2023. Automated Self-Supervised Learning for Recommendation. In *ACM Web Conference (WWW)*. 992–1002.

[41] Lianghao Xia, Chao Huang, Jiao Shi, and Yong Xu. 2023. Graph-less Collaborative Filtering. In *ACM Web Conference (WWW)*. 17–27.

[42] Han Xiao, Minlie Huang, Yu Hao, et al. 2015. TransA: An adaptive approach for knowledge graph embedding. *arXiv preprint arXiv:1509.05490* (2015).

[43] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. 2023. Debiased Contrastive Learning for Sequential Recommendation. In *The Web Conference (WWW)*. 1063–1073.

[44] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge Graph Contrastive Learning for Recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 1434–1443.

[45] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *International Conference on Information & Knowledge Management (CIKM)*. 4321–4330.

[46] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, et al. 2021. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *The Web Conference (WWW)*. 413–424.

[47] 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 *International Conference on Research and Development in Information Retrieval (SIGIR)*. 1294–1303.

[48] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *International Conference on Web Search and Data Mining (WSDM)*. 283–292.

[49] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. 353–362.

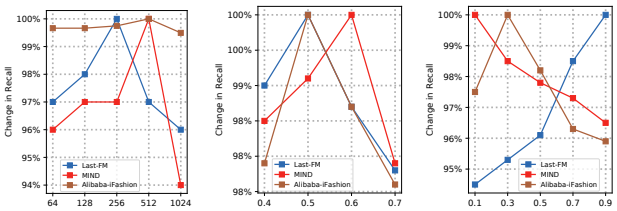
[50] Wayne Xin Zhao, Gaole He, Kunlin Yang, Hongjian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. Kb4rec: A data set for linking knowledge bases with recommender systems. *Data Intelligence* 1, 2 (2019), 121–136.

[51] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *International Conference on Information & Knowledge Management (CIKM)*. 1893–1902.

[52] 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 *International Conference on Research and Development in Information Retrieval (SIGIR)*.

# A APPENDIX

## A.1 Sensitivity to Key Hyperparameters




(a) Masking size $k_m$ (b) CL keep ratio $\rho$ (c) CL temperature $\tau$

**Figure 6: Hyperparameter Study of KGRec.**

In this study, we investigate the sensitivity of KGRec to changes in key hyperparameters, including the masking size $k_m$, the keep ratio for CL graph augmentation $\rho$, and the temperature for CL $\tau$. Our analysis reveal that the optimal hyperparameter settings are highly dependent on the characteristics of the underlying data. Specifically, we found that a masking size of 512 is ideal for MIND and Alibaba-iFashion, while 256 is optimal for Last-FM. Moreover, a CL keep ratio of 0.5 is the best choice for Last-FM and Alibaba-iFashion, while a temperature of 0.1 is recommended for MIND, 0.3 for Alibaba-iFashion, and 0.9 for Last-FM. We hypothesize that this difference in optimal temperature is due to the sparsity of the datasets, with denser datasets requiring higher temperatures to avoid false-negative samples. We suggest tuning the masking size and CL keep ratio in the ranges of $[128, 512]$ and $[0.4, 0.6]$, respectively, as a good starting point for tuning hyperparameters in other datasets. Although KGRec is relatively robust to small changes in hyperparameters, selecting the optimal settings is still critical for achieving the best performance.

## A.2 Explainability Study

In this section, we examine the interpretability of KGRec's recommendation results through case studies on knowledge rationalization. Specifically, we group news items in the MIND dataset by their preset categories and obtain the learned knowledge rationale scores for triplets connected to items within the same category. To provide an interpretable perspective, we calculate the average of rationale scores by triplet sets of the same relation $r$ and present the cases in Table 4. We select cases from five popular news categories, namely *sports*, *newspolitics*, *travel*, *finance*, and *tv-celebrity*. For each category, we showcase two of the relations with the highest average global rationale scores of their associated triplets. Our analysis reveals that KGRec is capable of effectively capturing the impact of user interests on the KG as rationales.

For instance, in the realm of sports news, users tend to focus on league categories and specific teams, and as such, these two types of relations in the knowledge graph are rationalized by the labels of user preferences. Similarly, the case of *newspolitics* demonstrates that users' political news preferences often have a strong partisan orientation, and they are also concerned with the positions of political figures. These examples highlight the explainability of our KGRec design. By explicitly modeling the hierarchical rationality in the knowledge graph, our approach can differentiate task rationales that reflect user interests. Moreover, the masking-reconstructing mechanism and cross-view rationale contrastive learning techniques help to emphasize and strengthen the rationale connections. This not only enhances the model's interpretability but also improves its performance by leveraging user preferences to make more accurate predictions. In summary, the rationalized knowledge graph and the KGRec architecture provide a robust framework for personalized recommendation that considers user preferences and interests in a structured and transparent manner.