CRCC: Collaborative Relation Context Consistency on the Knowledge Graph for Recommender Systems

Li-e Wang*†‡(✉), Huachang Zeng*, Shenghan Li*, Xianxian Li*†‡, Shengda Zhuo§, Jiahua Xie¶, Bin Qu*, Zhigang Sun*†, and Tianran Liu*

* School of Computer Science and Engineering, Guangxi Normal University, Guilin, China

† Guangxi Key Lab of Multi-source Information Mining Security, Guangxi Normal University, Guilin, China

‡ Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin, China

§ College of Cyber Security of Jinan University, Jinan University, Guangzhou, China

¶ School of Information Science and Engineering, Yunnan University, Kunming, China

{wanglie, lixx}@gxnu.edu.cn, {zhc28015, zhuosd96}@gmail.com, {lshcc9, xjhEde, qubin525, szung, ltinran}@163.com

***Abstract*—Knowledge graph (KG) as auxiliary information can solve the cold-start and data sparsity problems of recommender systems. However, most existing KG-based recommendation methods focus on how to effectively encode items with that users have interacted into entities and propagate them explicitly, but neglect the relation-level and context-level modeling of collaborative signals. Therefore, it is inevitable to incorporate some unrelated entities while utilizing a propagation strategy, which may weaken part of the recommendation performance.**

To address this problem, we propose a novel method named *Collaborative Relation Context Consistency* (CRCC). Compared with other KG-based methods, we model the relation-level and context-level of collaborative signals in a fine-grained manner. Specifically, we segment the user's collaborative knowledge graph to learn related entity information separately to enrich the embedding of users. Moreover, CRCC links the consistency score between the items that users and neighbors have interacted with as the fusion basis, and then we consider the inherent popularity of items while incorporating consistent entities to enhance the embedding representation of items. Extensive experiments on three real-world datasets show that CRCC outperforms several compelling baselines in both CTR prediction and top-K recommendation.

***Index Terms*—recommender systems, knowledge graph, collaborative relation, contextual knowledge**

I. Introduction

Recommender systems can help users find items of interest among massive amounts of information when their demands are uncertain. It builds a user preference model using machine learning technology and makes tailored recommendations to users [1].

Existing recommendation methods can be roughly categorized into three types: *collaborative filtering* (CF) [2], content-based [3], and hybrid [4]. Collaborative filtering has the problems of data sparsity and cold-start, which are typically resolved by introducing some auxiliary information, such as social relations [5] or KG [6]. KG is chosen as auxiliary information since it can improve recommendation interpretability.

KG is a directed heterogeneous graph in which nodes represent entities and edges represent relationships between entities. Extensive research shows that incorporating KGs into recommendation tasks improves performance [7]–[9]. Moreover, RippleNet [10] explored the potential interests of users along the entity-relation-entity link. CKAN [11] utilized the user's interactive items as propagation sources, obtaining the user's potential preference and the item's latent embedding by propagating multi-hop entities. Even though these models can iteratively propagate entities to facilitate recommendation, existing KG-based recommendation neglects the problem of collaborative relation context inconsistency due to the lack of comprehensive analysis of item type and fondness. Specifically, it can be divided into the two levels listed below:

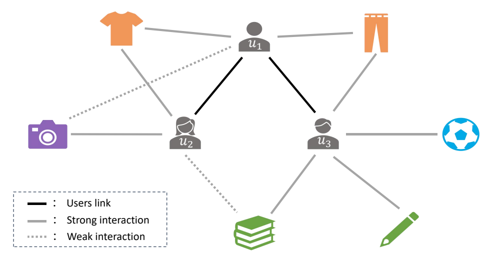


Fig. 1. A toy example reflecting the collaborative relation context inconsistency.

- The first is the context level, which indicates that items' context semantics among users in the user-item interaction graph may be different. An example of the inconsistency at the context level of items is shown in Fig. 1. Different colors represent different types of items, we can observe that u_3 will be an inconsistent neighbor of u_1. Because u_3 has interacted with item types other than clothes, such as sports and books (which may reflect greater interests), whereas u_1 interests appear to be only limited to clothes. As a result, they have quite a part of different item contexts.
- The second is the relation level. Previous work has usually focused on interactive and non-interactive relationships in user-item interaction scenarios. We make a more fine-grained distinction between interaction intentions, which are strong interactions with high rating values and

DOI reference number: 10.18293/SEKE2023-190

weak interactions with low rating values, to reveal how much the user likes the item. As shown in Fig. 1, we use the solid and dashed lines to indicate the strong and weak interaction relationships, respectively. We can see that users u_1 and u_2 are neighbors and that they are both connected to the camera. However, u_2 prefers the camera (reflecting a strong interaction relationship), while u_1 is not as interested in it (a weak interaction relationship). This leads to inconsistency at the relation level because although they are neighbors and are connected to the same item, their preferences for it are not consistent.

To address these above limitations, we propose an end-to-end model named CRCC, short for *Collaborative Relation Context Consistency*. To enrich the user's embedding, we segment the user's collaborative knowledge graph into a series of sub-views and learn the related entity information separately. Moreover, CRCC considers the relation-level and context-level of user-item interaction by quantifying the consistency between the items that users and neighbors have interacted with as the fusion basis. Next, we enhance the item embedding representation by considering the item's inherent popularity while diffusing entities with which the consistent users interacted.

Our contributions in this paper are summarized as follows:

- We propose knowledge feature learning to explore how users' interests change with the attractiveness of inter-entity relationships.
- To the best of our knowledge, we are the first work to address the inconsistency of collaborative relation context on KG-based recommendation.
- We consider the item's inherent popularity while diffusing entities with which the consistent users interacted.
- Extensive experiments on three public datasets show that CRCC over several convincing baselines.

II. Related Work

Existing KG-based recommendation methods can be classified into three categories:

- **Embedding-based** methods used *knowledge graph embedding* (KGE) [12] to map entities and relations to a low-dimensional vector space. CKE [13] adopted TransR to consider the heterogeneity of nodes and relationships to extract the structural representation of items. KTUP [8] proposed a multi-task learning model that adopted TransH for recommendation tasks and knowledge graph completion. However, these models learn entity embeddings are insufficient, making them more suitable for intra-graph tasks such as link prediction.
- **Path-based** methods enriched user-item interactions by designing connection paths between entities. RuleRec [14] utilized associations between items in KG for delivering an explainable recommendation. KPRN [7] generated path representations by integrating entity and relation semantics. However, designing a meta path manually is time-consuming and laborious, especially in extremely complex knowledge graphs.
- **Unified** methods combined the embedding-based and path-based methods to propagate embedding. KGCN [15] and KGNN-LS [16] demonstrated that aggregating entity neighbor information can improve recommendation performance. KGAT [17] proposed a collaborative knowledge graph that refines nodes' embeddings by propagating neighbor embeddings. CKAN [11] used a heterogeneous propagation strategy to encode diverse information for a better recommendation. However, existing unified methods focus on propagating the embedding of entities, ignoring the fine-grained analysis of the relational and contextual semantics from collaborative signals.

III. Problem Formulation

In this section, we formulate the KG-based recommendation problem as follows. In a typical recommendation scenario, we denote the sets of M users and N items by $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, respectively. The user-item interaction matrix $\mathbf{Y} \in \mathbb{R}^{M \times N}$ is determined based on the user's implicit feedback. $y_{uv} = 1$ indicates that user *u* has interacted with item *v*, otherwise $y_{uv} = 0$. In addition, $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$ denotes the knowledge graph, where *h*, *t* represent the head entity and the tail entity, respectively. *r* represents the relationship between the head entity *h* and the tail entity *t*; $\mathcal{E}$ and $\mathcal{R}$ represent the set of entities and relations, respectively. $\mathcal{A} = \{(v, \mathrm{e}) \mid v \in \mathcal{V}, \mathrm{e} \in \mathcal{E}\}$, where (v, e) indicates that item *v* can be aligned with entity *e*.

Given the user-item interaction matrix $\mathbf{Y}$ and the knowledge graph $\mathcal{G}$, our goal is to learn a prediction function $\hat{y}_{uv} = \mathcal{F}(u, v \mid \Theta, \mathcal{G})$, where $\hat{y}_{uv}$ denotes the probability that the user *u* will interact with items he has not engaged with before, and Θ denotes the model parameters.

IV. Methodology

The framework of CRCC is shown in Fig. 2, and then we elaborate on each module individually.

A. Knowledge Feature Learning

We treat the user's collaborative knowledge graph as a global view, and knowledge feature learning enriches the user preference representation by aggregating extended information containing sub-views of user interaction, which consists of knowledge feature segment and knowledge feature attentive.

1) Knowledge feature segment: We segment the global view formed by user *u* and his interactive entity set [i] into a series of sub-views based on the user preference profiles. The user's initial entity set is derived from item-entity alignment:

$$\varepsilon_u = \{e \mid (v, e) \in \mathcal{A} \; and \; v \in \{v \mid y_{uv} = 1\}\} \tag{1}$$

When combined with the segmented view, Eq.(1) can also be defined as follows:

$$\varepsilon_u^k = \left\{t \mid (h, r, t) \in \mathcal{G} \; and \; h \in \varepsilon_u^{k-1}\right\}, \quad k = 1, 2, \ldots, K \tag{2}$$

where *k* indicates the *k*-th sub-view being segment, we then define the *k*-th sub-view triple set for user *u* as follows:

$$S_u^k = \left\{(h, r, t) \mid (h, r, t) \in \mathcal{G} \; and \; h \in \varepsilon_u^{k-1}\right\}, \quad k = 1, 2, \ldots, K \tag{3}$$

[i]It consists of entities that the user interacted with and neighboring entities.

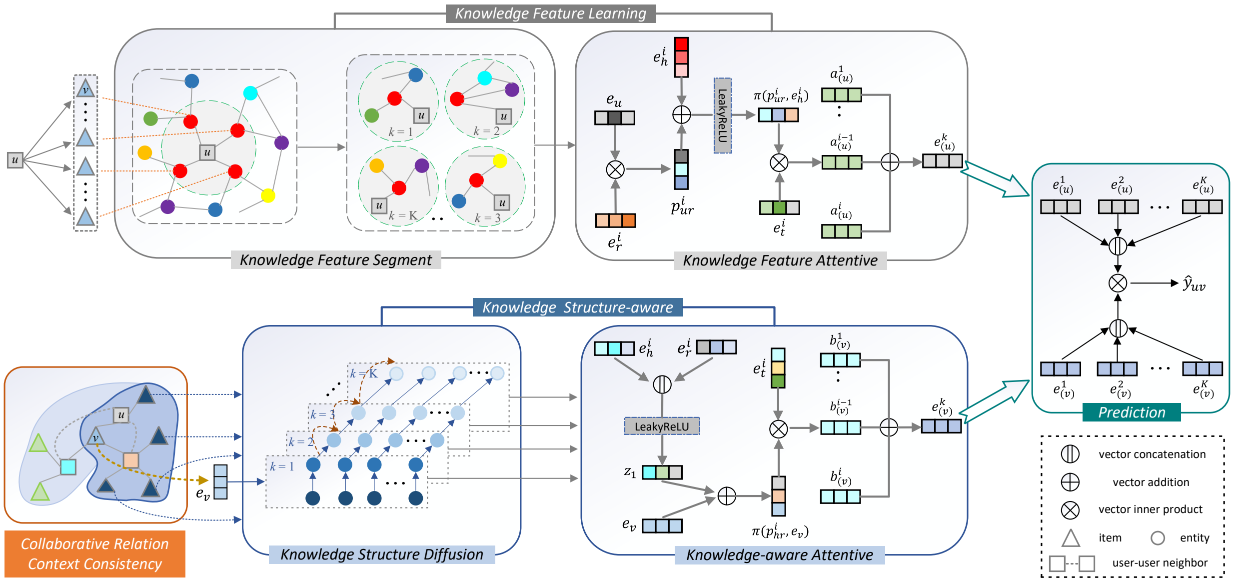


Fig. 2. Illustration of the proposed CRCC model, which consists of four modules: knowledge feature learning, collaborative relation context consistency, knowledge structure-aware, and prediction. Best view in color.

2) Knowledge feature attentive: We analyze the user's relation type preference for entities in the interaction set more thoroughly, and then try to incorporate more related entities for the relation type the user prefers:

$$p_{ur}^{i} = dynamic\_fun\left(e_u, e_r^i\right) \tag{4}$$

$$z_0 = \text{LeakyReLU}\left(W_0\left(p_{ur}^i + e_h^i\right) + b_0\right) \tag{5}$$

$$\pi\left(p_{ur}^i, e_h^i\right) = \sigma\left(W_2 \,\text{LeakyReLU}\left(W_1 z_0 + b_1\right) + b_2\right) \tag{6}$$

where e_u is the embedding of user u, e_r^i and e_h^i are the embedding of relation r and head entity h for the i-th triple, respectively. $W_* \in R^{2d\times d}$ is the trainable weight matrices, b_* is the biase terms of the neural network. $dynamic\_fun(\ ,\ )$ is a inner product function. p_{ur}^i characterizes the influence degree of the relation r between entities on the user u. Hereafter, we use the softmax function to normalize the coefficients across the whole triples in the triple set:

$$\pi\left(p_{ur}^i, e_h^i\right) = \frac{\exp\left(\pi\left(p_{ur}^i, e_h^i\right)\right)}{\sum_{(h',r',t')\in S_u^k}\exp\left(\pi\left(p_{ur'}^i, e_{h'}^i\right)\right)} \tag{7}$$

where $\pi\left(p_{ur}^i, e_h^i\right)$ controls the attentive weight generated from the user's relation preferences and the head entity. On the user side, we build the tail entity attentive embedding $a_{(u)}^i$ as follows:

$$a_{(u)}^i = \pi\left(p_{ur}^i, e_h^i\right) e_t^i \tag{8}$$

where e_t^i is the embedding of tail entity t for the i-th triple.

Finally, we obtain a representation of the k-th sub-view triple set for the user:

$$e_{(u)}^k = \sum_{i=1}^{\left|S_u^k\right|} a_{(u)}^i, \quad k = 1, 2, \ldots, K \tag{9}$$

where $\left|S_u^k\right|$ is the number of triples in set S_u^k.

B. Collaborative Relation Context Consistency

Distinct from previous methods that propagated users' interactive data layer by layer to obtain latent preferences. We consider both the relation of items that users have interacted with and the consistency of their contextual semantics. Taking user u as an example, neighbors with whom user u has interacted with item v are denoted as $S(u)$, all items that user u has interacted with are defined as $R(u)$. For user u and his interactive item v, it generates a semantic query embedding α_{uv} by mapping the concatenation of user u and item v embeddings:

$$q_{uv} = \sigma\left(W_a\left(e_u \| e_v\right)\right) \tag{10}$$

where e_u and e_v are the embedding of user u and item v, respectively. $\|$ denotes concatenation. Furthermore, we use the self-attention mechanism to calculate the user u's degree of preference α_{uv} for the interactive item as follows:

$$\alpha_{uv} = \frac{\exp\left(W_b\left(q_{uv}\|e_{r_{uv}}\right)\right)}{\sum_{j\in R(u)}\exp\left(W_b\left(q_{uj}\|e_{r_{uj}}\right)\right)} \tag{11}$$

where $e_{r_{uv}}$ is the relation embedding for user u interactive item v. Analogously, for neighbor user p and his interactive items, we obtain a degree of preference β_{pv} for user p in the same way. Next, we consider the consistency between the user and his neighbors and the items with that they have interacted. The final consistency score γ_{up} is defined as follows:

$$\gamma_{up} = W_d \cdot \sigma\left(W_c \cdot fusion\left(\alpha_{uv}e_{r_{uv}}, \beta_{pv}e_{r_{pv}}\right) + b_c\right) + b_d \tag{12}$$

$$\gamma_{up} = \frac{\exp\left(\gamma_{up}\right)}{\sum_{p'\in S(u)}\exp\left(\gamma_{up'}\right)} \tag{13}$$

where $fusion(\ ,\ )$ is a inner product function. Normalizing Eq.(12) by the softmax function yields the consistency score between the two users. We carefully tune this consistency score γ_{up} at a reasonable threshold and find that a threshold higher than about 0.6 is better for filtering out users with high consistency scores, which also means that some of the noise caused by aggregating entities to multiple orders is diminished. $\mathcal{U}_{consis}$ will be denoted as the set of consistent users.

C. Knowledge Structure-aware

Different users express various degrees of fondness for an item based on its inherent popularity. Knowledge structure-aware enhances item embedding by accounting for the item's inherent popularity while also improving the representation of consistent users' interactive entities, which consists of knowledge structure diffusion and knowledge-aware attentive.

1) Knowledge structure diffusion: It considers the inherent popularity of an item while iteratively diffusing user interactive entities. On the item side, the mapping of items interacted by $\mathcal{U}_{consis}$ with a high consistency score into entities is defined as follows:

$$V_u = \{v_u \mid u \in \{u \mid y_{uv} = 1\} \text{ and } u \in \mathcal{U}_{consis}\} \tag{14}$$

$$\varepsilon_v = \{e \mid (v_u, e) \in \mathcal{A} \text{ and } v_u \in V_u\} \tag{15}$$

When combined with the diffusion order, the above Eq.(15) can also be defined equally as follows:

$$\varepsilon_v^k = \left\{t \mid (h, r, t) \in \mathcal{G} \text{ and } h \in \varepsilon_v^{k-1}\right\}, \quad k = 1, 2, \ldots, K \tag{16}$$

where *k* indicates the distance from the initial entity set. Given the definition of entity set, we then define the *k*-th order triple set for item *v* as follows:

$$S_v^k = \left\{(h, r, t) \mid (h, r, t) \in \mathcal{G} \text{ and } h \in \varepsilon_v^{k-1}\right\}, \quad k = 1, 2, \ldots, K \tag{17}$$

2) Knowledge-aware attentive: Given an entity set for consistent users interaction, we consider the role of the item's inherent popularity while maintaining the connection between the head entity and the relation, which is formulated as follows:

$$z_1 = \operatorname{LeakyReLU}\left(W_0\left(e_h^i \| e_r^i\right) + b_0\right) \tag{18}$$

$$\pi\left(p_{hr}^i, e_v\right) = W_2^T \cdot \sigma\left(W_1 \cdot pop\_fun\left(z_1, e_v\right) + b_1\right) + b_2 \tag{19}$$

where z_1 denotes the semantics representation of the head entity and relation aggregation. $pop\_fun(\ ,\ )$ is a function, and the addition is found with the best performance in the experiments. Hereafter, we normalize the coefficients across the whole triples in the triple set by adopting the softmax function:

$$\pi\left(p_{hr}^i, e_v\right) = \frac{\exp\left(\pi\left(p_{hr}^i, e_v\right)\right)}{\sum_{(h', r', t') \in S_v^k} \exp\left(\pi\left(p_{h'r'}^i, e_v\right)\right)} \tag{20}$$

where $\pi\left(p_{hr}^i, e_v\right)$ controls the attentive weight generated from aggregated semantics of head entity and relation and item's inherent popularity. On the item side, we build the tail entity attentive embedding $b_{(v)}^i$ as follows:

$$b_{(v)}^i = \pi\left(p_{hr}^i, e_v\right) e_t^i \tag{21}$$

Finally, we obtain a representation of the *k*-th order triple set for item:

$$e_{(v)}^k = \sum_{i=1}^{\left|S_v^k\right|} b_{(v)}^i, \quad k = 1, 2, \ldots, K \tag{22}$$

where $\left|S_v^k\right|$ is the number of triples in set S_v^k.

D. Model Prediction and Loss Function

After segmenting into K sub-views and iterating the K-order aware process, we obtain the embedding set of user and item with $e_{(u)}^k$ and $e_{(v)}^k$ for $k = [1, 2, \ldots, K]$. And then for each user *u*, his final embedding is denoted as: $e_{(u)}^* = \left[e_{(u)}^1 \| e_{(u)}^2 \| \ldots \| e_{(u)}^K\right]$ that concatenates his embedding at each sub-view. Similarly, each item *v* final embedding is: $e_{(v)}^* = \left[e_{(v)}^1 \| e_{(v)}^2 \| \ldots \| e_{(v)}^K\right]$. Finally, the predicted rating is represented by the inner product of the final user and item embeddings:

$$\hat{y}_{uv} = e_{(u)}^* \cdot e_{(v)}^* \tag{23}$$

For each user, we extract the same number of negative samples as positive samples. Afterward, we have the following loss function for CRCC:

$$\mathcal{L} = \sum_{(u, v, j) \in Y} -\ln \sigma\left(\hat{y}_{uv} - \hat{y}_{uj}\right) + \lambda \|\Theta\|_2^2 \tag{24}$$

where $Y = \{(u, v, j) \mid (u, v) \in Y^+, (u, j) \in Y^-\}$, $\sigma(x)$ is the sigmoid function, Θ is the model parameters set, and $\|\Theta\|_2^2$ is the $L2$-regularizer that parameterized by λ.

V. EXPERIMENT

A. Experiment Settings

1) Datasets and Evaluation Metrics: We adopt three benchmark datasets related to Microsoft KG Satori. The detailed statistics are summarized in Table I. We employ two widely used metrics *AUC* and *F1* to evaluate the performance of CTR prediction, and then choose *Recall@K* to evaluate the effectiveness of top-K recommendation. Note that higher values of the three metrics indicate better performance.

2) Baselines: We compare CRCC with the following baselines. BPRMF [18] is a CF-based method that optimizes implicit feedback using pairwise matrix factorization. CKE [13] combines the CF module with textual, structural, and visual knowledge embeddings in a unified framework. PER [19] is a path-based method that treats the KG as a heterogeneous information network and extracts meta-path based on features. RippleNet [10] propagates users' potential preferences in the KG to enrich user representations. KGCN [15] is the first work to integrate GCN to KG-based recommendation, which iteratively aggregates information about neighboring entities. KGNN-LS [16] transforms a heterogeneous knowledge graph into a user-specific weighted graph. KGAT [17] uses an attention mechanism to prioritize neighbors in collaborative knowledge graphs. CKAN [11] propagates user interactive data layer by layer to high-order entities to learn user preferences and item embedding. KGIN [20] decouples user-item interactions at the granularity of user intents and implements GNN to the proposed user-intent-item-entity graph.

3) Parameter Settings: In CRCC, we divide each dataset into training, validation, and test sets with the proportion of 6:2:2. The embedding size is fixed at 64, the learning rate is set as 0.002, and the coefficient of *L*2 normalization is tuned as 0.00001. The size of the user's triple set is set to 8 or 16, and then the size of the item's triple set is fixed at 64. The batch size is fixed at 2048. We set the epoch on all three real-world datasets to 20, each experiment is repeated five times, and the average performance is reported.

B. Overall Comparison

Table II and Fig. 3 show that CRCC has obvious advantages over existing state-of-the-art baselines. Analyzing such performance comparison, we have the following observations:

Table I. Statistics of the three real-world datasets.

	Last.FM	Book-Crossing	MovieLens-1M
# users	1872	17 860	6036
# items	3846	14 967	2445
# interactions	42 346	139 746	753 772
# entities	9366	77 903	182 011
# relations	60	25	12
# triples	15 518	151 500	1 241 996

Table II. The result of *AUC* and *F1* in CTR prediction.

Model	Last.FM		Book-Crossing		MovieLens-1M	
	AUC	F1	AUC	F1	AUC	F1
BPRMF	0.756	0.701	0.658	0.611	0.892	0.792
PER	0.641	0.603	0.605	0.572	0.712	0.667
CKE	0.747	0.674	0.676	0.623	0.907	0.802
RippleNet	0.776	0.702	0.721	0.647	0.918	0.842
KGCN	0.796	0.721	0.684	0.631	0.909	0.837
KGNN-LS	0.805	0.722	0.676	0.631	0.914	0.841
KGAT	0.829	0.742	0.731	0.654	0.914	0.844
CKAN	0.842	<u>0.769</u>	<u>0.753</u>	<u>0.673</u>	0.915	<u>0.845</u>
KGIN	<u>0.849</u>	0.760	0.727	0.661	<u>0.919</u>	0.844
CRCC	**0.859**	**0.778**	**0.766**	**0.681**	**0.931**	**0.862**

- CRCC consistently outperforms all baselines in all metrics across three datasets. Specifically, CRCC improves over the state-of-the-art baselines w.r.t. *AUC* by 1.2%, 1.7% and 1.3% in Last.FM, Book-Crossing and MovieLens-1M, respectively.
- Compared with CKAN and KGIN, the performance of CRCC justifies the effectiveness of the consistency of collaborative relation context. The results compared with KGCN and KGNN-LS show the significance of explicitly encoding collaborative signals.
- Comparing BPRMF with CKE, using KG significantly improves the performance of matrix factorization. This finding is also reflected in KTUP.
- The efficiency of a model is dependent on how it uses the KG information. The CF-based matrix factorization method BPRMF outperforms the path-based model PER, perhaps since it's not always possible to find the optimal meta-path. The unified approach outperforms embedding-based and path-based baselines.

C. Ablation Study

To evaluate the efficacy of each component in our proposal, we compare CRCC with three variants and Table III shows the experimental results.

- CRCC $_{w/o\ K\&F}$: This variant is CRCC without the knowledge feature learning component.
- CRCC $_{w/o\ R\&C}$: This variant is CRCC without the collaborative relation context consistency component.
- CRCC $_{w/o\ K\&S}$: This variant is CRCC without the knowledge structure-aware component.

The following conclusions can be drawn from an analysis of the data in Table III: (1) CRCC $_{w/o\ K\&F}$ can impair part of the performance, which demonstrates the utility of segmenting a user's collaborative knowledge graph into a series of sub-views and learning related entity information separately. (2) The performance of recommendations is significantly reduced by CRCC $_{w/o\ R\&C}$, which emphasizes the importance of sampling high consistency score data between the user and his neighbors and their interacted items. (3) CRCC $_{w/o\ K\&S}$ can substantially worsen recommendation performance, which is especially apparent on Last.FM and Book-Crossing. This demonstrates the benefits of considering the item's inherent popularity while maintaining consistency in the diffusion entities with which users have interacted. Overall, it is clear that CRCC consistently achieves the best results.

Table III. The result of *AUC* w.r.t effects different of components.

Category	Last.FM	Book-Crossing	MovieLens-1M
CRCC $_{w/o\ K\&F}$	0.844	0.740	0.920
CRCC $_{w/o\ R\&C}$	0.825	0.733	0.913
CRCC $_{w/o\ K\&S}$	0.838	0.744	0.919
CRCC	**0.859**	**0.766**	**0.931**

Table IV. The result of *AUC* w.r.t different dimension of embedding.

d	4	8	16	32	64	128
Last.FM	0.798	0.823	0.837	0.843	**0.859**	0.833
Book-Crossing	0.689	0.734	0.735	0.739	0.743	**0.766**
MovieLens-1M	0.884	0.887	0.902	0.907	**0.931**	0.913

Table V. The result of *AUC* w.r.t different numbers of sub-view and order (k).

k	Last.FM	Book-Crossing	MovieLens-1M
1	0.838	0.736	0.913
2	0.845	0.746	**0.931**
3	**0.859**	**0.766**	0.920
4	0.849	0.753	0.919

Table VI. The result of *AUC* on MovieLens-1M w.r.t different sizes of triple set.

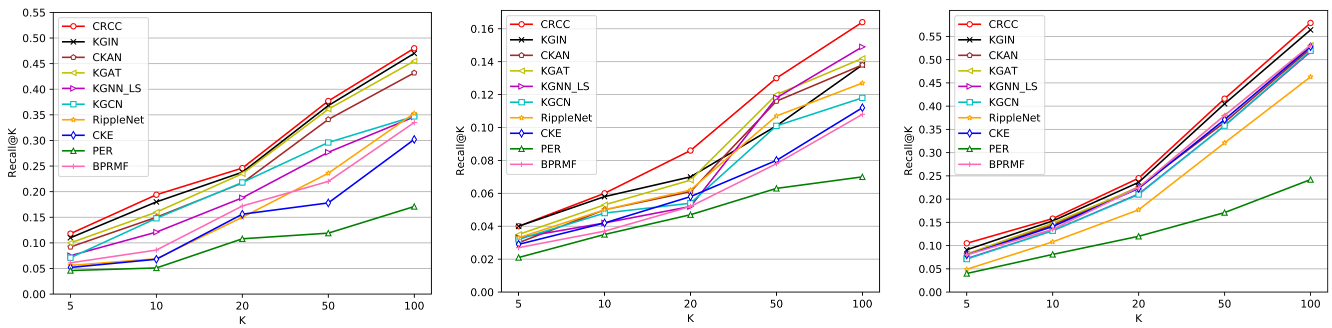
user \ item	4	8	16	32	64
4	0.901	0.901	0.900	0.901	0.901
8	0.908	0.908	0.908	0.908	0.904
16	0.914	0.914	0.914	0.914	**0.931**
32	0.918	0.918	0.917	0.917	0.918
64	0.920	0.919	0.919	0.919	0.919

D. Sensitivity Analysis

1) Impact of Dimension of Embedding: Table IV shows that increasing the embedding dimension improves the performance of CRCC within a certain range. But CRCC performance would decline with excessive dimension. This is due to the fact that when the embedding dimension increases, more information is encoded into it, but it also causes a little overfitting problem.

2) Impact of Numbers of Sub-view and Order: As shown in Table V, the best performance is achieved when k is 3, 3 and 2 in Last.FM, Book-Crossing and MovieLens-1M, respectively. This may be due to a trade-off between the number of segmented views and the order of knowledge diffusion. Too few segmented views and knowledge diffusion order are insufficient to capture entity relationships, but too large numbers may contain irrelevant noise information.

3) Impact of Size of Triple Set: Table VI demonstrates that the optimal size of the item triple set for MovieLens-



(a) Last.FM (b) Book-Crossing (c) MovieLens-1M

Fig. 3. The result of *Recall@K* in top-K recommendation.

1M is 64, which confirms that increasing the size can include more related entity sets and improve the recommendation performance. The best performance is achieved when the user triple set is set to 16. When it surpasses a reasonable threshold, the recommendation result decreases, thus we need a suitable size of user triple set.

VI. CONCLUSION

In this paper, we focus on the CRCC, which links the consistency score between the items that users and neighbors have interacted with as the fusion basis. This is the first work to address the inconsistency of collaborative relation context on KG-based recommendation. Extensive experiments on three real-world datasets demonstrate the effectiveness of CRCC. In the future, we will emphasize the effective utilization of multi-modal learning on collaborative knowledge graphs for better recommendation performance.

ACKNOWLEDGMENT

This work is supported in part by the National Natural Science Foundation of China (Nos.62262003 and U21A20474), the Guangxi Science and technology project (Guike AA22067070), the Guangxi Natural Science Foundation (No.2020GXNSFAA297075), the Center for Applied Mathematics of Guangxi (Guangxi Normal University), the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, the Guangxi Collaborative Innovation Center of Multi-source Information Integration and intelligent Processing, the Guangxi Talent Highland project of Big Data Intelligence and Application.

REFERENCES

[1] Yankai Chen, Yaming Yang, Yujing Wang, Jing Bai, Xiangchen Song, and Irwin King, "Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation," in *ICDE*. IEEE, 2022, pp. 299–311.

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

[3] Mengqi Liao, S Shyam Sundar, and Joseph B. Walther, "User trust in recommendation systems: A comparison of content-based, collaborative and demographic filtering," in *CHI*, 2022, pp. 1–14.

[4] Xiaoli Tang, Tengyun Wang, Haizhi Yang, and Hengjie Song, "Akupm: Attention-enhanced knowledge-aware user preference model for recommendation," in *SIGKDD*, 2019, pp. 1891–1899.

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

[6] Weiran Pan, Wei Wei, and Xian-Ling Mao, "Context-aware entity typing in knowledge graphs," *EMNLP*, 2021.

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

[8] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in *WWW*, 2019, pp. 151–161.

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

[10] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo, "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in *CIKM*, 2018, pp. 417–426.

[11] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu, "Ckan: collaborative knowledge-aware attentive network for recommender systems," in *SIGIR*, 2020, pp. 219–228.

[12] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *AAAI*, 2015, vol. 29.

[13] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma, "Collaborative knowledge base embedding for recommender systems," in *SIGKDD*, 2016, pp. 353–362.

[14] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren, "Jointly learning explainable rules for recommendation with knowledge graph," in *WWW*, 2019, pp. 1210–1221.

[15] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo, "Knowledge graph convolutional networks for recommender systems," in *WWW*, 2019, pp. 3307–3313.

[16] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang, "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," in *SIGKDD*, 2019, pp. 968–977.

[17] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua, "Kgat: Knowledge graph attention network for recommendation," in *SIGKDD*, 2019, pp. 950–958.

[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *UAI*, 2012.

[19] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han, "Personalized entity recommendation: A heterogeneous information network approach," in *WSDM*, 2014, pp. 283–292.

[20] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua, "Learning intents behind interactions with knowledge graph for recommendation," in *WWW*, 2021, pp. 878–887.