Highlights

Items Proxy Bridging: Enabling Frictionless Critiquing in Knowledge Graph Recommendations

Huanyu Zhang, Xiaoxuan Shen, Yu Lei, Baolin Yi, Jianfang Liu, Yinao xie

- A new generic critiquing framework (IPGC) is proposed.
- IPGC can work on mostly CF knowledge graph recommendation algorithms.
- Novel items proxy mechanism was designed to adapt to Collaborative Filtering.
- IPGC effectively mitigates the catastrophic forgetting problem of multi-critiquing.
- IPGC outperforms baselines on sota KG recommender models and benchmark datasets.

Items Proxy Bridging: Enabling Frictionless Critiquing in Knowledge Graph Recommendations*

Huanyu Zhang[a,1], Xiaoxuan Shen[a], Yu Lei[a], Baolin Yi[a,*], Jianfang Liu[a] and Yinao xie[a]

[a]Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan, 430079, Hubei, China

ARTICLE INFO

Keywords:
Critiquing
Recommendation
Collaborative Filtering
Knowledge Graph

ABSTRACT

Modern recommender systems place great inclination towards facilitating user experience, as more applications enabling users to critique and then refine recommendations immediately. Considering the real-time requirements, critique-able recommender systems typically straight modify the model parameters and update the recommend list through analyzing the user critiquing keyphrases in the inference phase. Current critiquing methods require first constructing a specially designated model which establish direct correlations between users and keyphrases during the training phase allowing for innovative recommendations upon the critiquing, restricting the applicable scenarios. Additionally, all these approaches ignore the catastrophic forgetting problem, where the cumulative changes in parameters during continuous multi-step critiquing may lead to a collapse in model performance. Thus, We conceptualize a proxy bridging users and keyphrases, proposing a streamlined yet potent *Items Proxy Generic Critiquing Framework* (IPGC) framework, which can serve as a universal plugin for most knowledge graph recommender models based on collaborative filtering (CF) strategies. IPGC provides a new paradigm for frictionless integration of critique mechanisms to enable iterative recommendation refinement in mainstream recommendation scenarios. IPGC describes the items proxy mechanism for transforming the critiquing optimization objective of user-keyphrase pairs into user-item pairs, adapting it for general CF recommender models without the necessity of specifically designed user-keyphrase correlation module. Furthermore, an anti-forgetting regularizer is introduced in order to efficiently mitigate the catastrophic forgetting problem of the model as a prior for critiquing optimization. Extensive experimental results indicate the IPGC outperforms the state-of-the-art critiquing methods under various types of critiques. Meanwhile, the results show that IPGC can effectively operate on original model with different architectures and alleviate the catastrophic forgetting problem significantly.

1. Introduction

Modern recommender systems are increasingly emphasizing the user experience while exploring methods to offer better recommendations [37, 26, 50]. More apps are allowing users to interact with straightforward feedback and deliver immediate, effective responses with refined recommendations, which can significantly elevate the user experience and perceptions. For example, as shown in Figure 1(a), users can continuously present suggestions to gain satisfied movies. In each interaction, the system analyzes the critiquing information and innovates recommendations, while the user could choose to accept or continue with other opinions. In Figure 1(b), the illustrated app proactively supplies users with various categories of keyphrases, allowing them to argue or prefer these keyphrases in order to precisely capture users' preferences.

Critique-able recommender systems typically involve the training phase of the original model and the inference phase in which the recommendations are progressively refined by analyzing the critiquing information. Recent works mostly focus on the study of conversational recommender system (CRS), allowing users to critiquing both items and keyphrases [17, 42, 18, 14, 44]. Considering that keyphrases can deliver more explicit and detailed information, researchers place more attention on the keyphrases. While these methods have proven to be capable of adapting recommendations from critiquing in CRS scenarios, there are two concerns remain to be addressed. Firstly, the previous researches have all designed specifically models that can directly model the correlation between users and keyphrases in the training phase for the benefit of accepting critiquing in the inference phase. These approaches, though enabling the model to handle critiquing, restrict the available scenarios and exhibit weak initial recommendation performance. Secondly, a consistent phenomenon present in the experimental results of these previous works is that the tail of multi-step critiquing always shows a trend of drastic performance decline, implying that they are not capable of handling the demands of more round interactions. This limitation stems from their shared neglect of inference-phase parameter drift in latent variables, where successive modifications induce progressive erosion of critical feature representations encoded during initial model training, ultimately manifesting as catastrophic forgetting [47, 1, 48].

Hence, the integration of critiquing in generalized applications via knowledge graphs shows measurable potential to relieve specific limitations characterizing current methods. Recent evidence [32, 38, 39] has indicated that knowledge graph-enhanced recommendation models are very powerful.

*Corresponding author

ccnuzhy@mails.ccnu.edu.cn (H. Zhang); shenxiaoxuan@ccnu.edu.cn (X. Shen); leiyu@mails.ccnu.edu.cn (Y. Lei); epower@mail.ccnu.edu.cn (B. Yi); jianfangliu@mails.ccnu.edu.cn (J. Liu); xieyinao@mails.ccnu.edu.cn (Y. xie)

ORCID(s): 0000-0003-1991-565X (H. Zhang)

Furthermore, knowledge graph (KG) could help critique-able recommender system to better understand user critiquing as a data structure containing rich knowledge and semantic information [19, 40, 28]. Nevertheless constructing a universal critique-able model based on the KG recommendation method can be difficult. Most of these efforts are designed on collaborative filtering (CF) strategy, only modeling the direct users' preferences on items, and without structuring the straightforward correlation between users and keyphrases. Thus, we need to design a method that is generalizable among CF recommendation models [43, 8] to represent the relationships between users and keyphrases. Additionally, the critical yet underexplored phenomenon of catastrophic forgetting in sequential critique scenarios poses significant risks, where progressive parameter drift during multi-step refinement processes may induce cascading performance degradation in recommendation systems. These observations necessitate the development of knowledge preservation mechanisms that maintain foundational knowledge representations throughout iterative critiquing cycles to mitigate catastrophic knowledge loss.

Consequently, we introduce a bayesian *Items Proxy Generic Critiquing Framework* (IPGC), that can operate as a plugin for most current major KG recommender systems relying on CF strategy, responding to user feedback in real-time and having the capability of anti-forgetting. For constructing the intrinsically absent user-keyphrase associations in generally CF models, we solve for these keyphrases by identifying proxy bridging to achieve the delivery of critiquing information, indirectly structure the correlations between users and keyphrases. This methodology enables zero-interference critiquing integration while preserving baseline recommendation integrity. Obviously, there definitely exists two optimization objectives in KG recommender model which are user's preference for items and the relationships between items and keyphrases, respectively. Using items as the proxy is thus a very rational selection. Moreover, the employment of the items proxy permits us to concentrate only on the direct connections between users and items during the inference phase, making it naturally adapted to the original CF model. Since different KG recommenders are mostly different in the manner of training KG structure, a generic method to match the relevance among items proxy and keyphrase is quite challenging to realize.We choose to leverage the most intuitive approach by treating all the items in KG linked to the keyphrase as its proxy and designing a multi-hop sampling protocol to ensure unbiased preference estimation across users' preferences for keyphrases. In addition, in IPGC we constructed a gradient-based anti-forgotten regularizer that labels essential parameters in user embedding prior and penalizes the changes of these key variables to ensure maximum prevention of the loss of critical information learned from the original model, thus alleviating the forgetting problem. In general, our contributions are presented as follows:

- We propose a generic IPGC framework that indirectly structures the correlation between user and keyphrase through items proxy bridging, which can frictionless effectively refine recommendations based on user critiquing. To the best of our knowledge, IPGC is the first critiquing framework with generalizability.
- We use a gradient-based anti-forgotten regularizer which can effectively prevent the model from forgetting important characteristics in the original model during the inference phase.
- We conducted extensive experiments on two real-world datasets with sota KG recommendation algorithms KGAT, KGIN, and DiffKG, the results demonstrate the effectiveness and generalizability of our IPGC framework.

2. Related Work

Knowledge Graph Recommendation has emerged as one of the most popular and well-performing models in the field of recommender systems, as they can enhance recommendations by utilizing the large amount of knowledge in KG. Early approaches like CKE [49] and DKN [33] leveraged pre-training on KG to strengthen item embeddings for improved recommendations [31]. In recent years, end-to-end models based on graph convolution [36, 38, 39] or graph attention networks [32, 34, 41, 5, 51] have proven to be more effective in mining information from KG, which are the which are state-of-the-art solutions that exhibit strong competitiveness. In addition, multimodal knowledge graph have been attempted to be applied [27], and there are also methods that try to reduce the noise from higher-order information [15]. Some methods [35] consider the coupling between KG representation task and the recommendation task or incorporate collaborative information [41], or try to mine the information more valuable to the recommendation task in KG. More recent schemes have incorporated graph contrastive learning on the basis of GNN to reduce the potential noise [54, 46, 55, 45, 11, 7], yielding better recommendation performance. Overall, the objectives of researchers mainly focus on mining semantic and higher-order information in kg more efficiently and accurately and better integrating knowledge graph representing with collaborative filtering recommendations. While KG recommendations are highly competitive and can provide users with explanations, the overhead of time and space makes it challenging to update recommendations immediately with user critiquing.

Critique-able Recommendation differs from traditional recommendation systems, which only need to model user preferences for items, whereas it also requires modeling the correlation between users and keyphrases [20, 44, 23, 3, 4, 2]. Recent work primarily focused on conversational recommender systems, gradually adapting recommendations to the practical demands of the user through iterative interactions between the user and the conversational system [53, 6, 9, 52]. However, most of these approaches, in order to model the association between users and keyphrases, are implemented within specially designed Variational Autoencoder (VAE)

frameworks [16, 24, 25]. They achieve critiquing by redesigning the VAE encoder to achieve the input of directly correlating users with keyphrases into the latent space. In continuous critiquing task, they modify the potential parameters directly [18, 20] or using Bayesian methods [44, 23] to update the user embedding. These methods can effectively incorporate critiquing to decode the refined recommendations. Although these methods indeed demonstrate the ability to tune recommendations from user feedback, they underutilize critiquing information. Furthermore, the adapted VAE models cannot be applied to traditional common recommendation scenarios nor handle large volumes of data; they are essentially and specifically designed for critique tasks. Additionally, to the best of our knowledge, BCIE [29] is the first attempt to combine KG with critiquing in CRS. It can effectively utilize the knowledge in KG, illustrating the potential of the KG for critiquing. Nonetheless, it remains an inherently recommender method dedicated to critiquing and is incapable of being employed in traditional list-based recommendation scenarios or state-of-the-art methods.

In conclusion, KG recommender algorithms have become one of the mainstream methods due to their performance advantages. And the utilization of KG enables better prehension of critiquing keyphrases. Therefore, we aim to establish a universal critique-able recommender paradigm based on KG recommendation models. This paradigm will seamlessly adapt to mostly CF methods in the KG recommendation scenario, patching the shortcomings of previous approaches and enabling rapid and efficient adaptation to user feedback. Not only overcome many of the shortcomings of previous approaches, but also integrates seamlessly with state-of-the-art methodologies.

3. METHODOLOGY

3.1. Problem Formulation

We first introduce the knowledge graph recommendation and formulate the critiquing task.

Knowledge Graph Recommendation. Given a user set $\mathcal{U}$, an item set $\mathcal{V}$, and a knowledge graph $\mathcal{G}$ containing a large number of semantic keyphrases $\mathcal{K}$ with the complex relationships between them. The objective of a knowledge graph recommender system is to predict the items that users probably interact with in the future, according to their historical interactions $\mathcal{D}_T = \{(u, v) | u \in \mathcal{U}, v \in \mathcal{V}\}$ and the high-order information provided by KG. Additionally, it can provide recommendation explanations $\mathcal{K}_e$ for users from KG.

Critiquing Task Formulation. In a critique-able recommender system, users can also express their recognition of the recommendations, they can readily accept or critique the explanation keyphrases $k \in \mathcal{K}_e$ to make the results more in line with their perceptions. Specifically, given the user embedding $\mathbf{u}$, item embedding $\mathbf{v}$ generated by the original KG model, we aim to achieve a universal iterative critiquing mode for general KG recommender systems. It allows the user to repeatedly interact with the system and generates an updated user embedding $\mathbf{u}$ with the user critiquing data $\mathcal{D}_c = \{(u, k) | u \in \mathcal{U}, k \in \mathcal{K}\}$ to renew scores for the candidate items and re-recommend top-k items.

3.2. General KG Recommender Model

In this section, we will introduce the basic principle of KG recommendation model, which is shown on the left side of Figure 2. Related work summarizes various types of KG recommendation models, which share the common idea of leveraging semantic and higher-order information among entities in the knowledge graph to explore potential interactions between users and items. These methods essentially optimize the representations of users, items, and keyphrases given a training set and knowledge graph. We represent the optimization objectives of the original knowledge graph model as follows:

$$\mathcal{L}_O = \log p\left(U, V, K | \mathcal{D}_T, \mathcal{G}\right) \tag{1}$$

where U, V, and K stand for user embeddings, item embeddings, and keyphrase embeddings, respectively. Specifically, these methods generally have two separate optimization objectives, namely optimizing user preferences for items:

$$\mathcal{L}_{REC} = \sum_{(u,v)\in\mathcal{D}_T} \sum_{(u,v')\notin\mathcal{D}_T} \log \mathcal{F}_{rec}\left(\hat{y}_{uv} - \hat{y}_{uv'}\right) \tag{2}$$

and aggregating high-order information from the knowledge graph, which means optimizing the structural information between items and keyphrases:

$$\mathcal{L}_{KG} = \sum_{(v,k)\in\mathcal{G}} \mathcal{F}_{kg}(v, k) \tag{3}$$

Besides, some methods have introduced other more complicated optimization objectives to achieve better recommendation performance with some results. Still, they revolve around the two optimization objectives mentioned above.

3.3. IPGC Framework

The proposed IPGC targets to refine the recommendations depending on the user critiquing $\mathcal{D}_c$. In fact, we merely update the user embedding and keep the item embedding and keyphrase embedding invariant. The updated user embedding U^* can be obtained from $p\left(U^* | \mathcal{D}_c, V, K\right)$, and according to the Bayesian formula we will have:

$$p\left(U^* | \mathcal{D}_c, V, K\right) \propto \underbrace{p\left(\mathcal{D}_c | U^*, V, K\right)}_{critiquing\ likelihood} \times \underbrace{p\left(U^*\right)}_{user\ prior} \tag{4}$$

Apply the log function to convert Eq.4 from multiplication to addition:

$$\log p\left(U^* | \mathcal{D}_c, V, K\right) \propto \log p\left(\mathcal{D}_c | U^*, V, K\right) + \log p\left(U^*\right) \tag{5}$$

More specifically, the goal is to optimize the user embedding posterior U^* by calculating the user potential preference for the keyphrase, and to preserve the user embedding prior

learned from the original model as much as possible. According to Maximum A Posteriori Estimation (MAP), the optimization objective of IPGC is to maximize the user embedding posterior estimate:

$$\begin{aligned} U^* &= \arg\max_{U^*} p\left(U^* | \mathcal{D}_c, V, K\right) \\ &= \arg\max_{U^*} p\left(\mathcal{D}_c | U^*, V, K\right) \times p\left(U^*\right) \\ &= \arg\max_{U^*} \left(\log p\left(\mathcal{D}_c | U^*, V, K\right) + \log p\left(U^*\right)\right) \end{aligned} \tag{6}$$

Firstly, for the critiquing likelihood, it can be expressed as the user preference for the keyphrase:

$$p\left(\mathcal{D}_c | U^*, V, K\right) = \prod_{\langle u_i, k_j \rangle \in \mathcal{D}_c} \left(p\left(\langle u_i, k_j \rangle\right)\right) \tag{7}$$

where $p\left(\langle u_i, k_j \rangle\right)$ denotes the preference probability of user u_i for keyphrase k_j, in this paper which will be represented by the BPR score [21]. Unfortunately, directly addressing $p\left(\langle u_i, k_j \rangle\right)$ is challenging. This is because, in KG recommender model, there is no direct correlation established between users and keyphrases, making it impossible to calculate the BPR score between user and keyphrase through feature cross-methods.

Although it is difficult to compute $p\left(\langle u_i, k_j \rangle\right)$ straight forward, the original model has already learned the preferences feature between users and items as well as the structural information between items and keyphrases in KG through $\mathcal{L}_{REC}$ and $\mathcal{L}_{KG}$, respectively. Considering the item proxy as a critical node for transmitting preference information can implicitly calculate the BPR score between users and keyphrases. Hence, IPGC describes a universal framework for indirectly computing the preference of users for keyphrases with item proxy. Formally, critiquing likelihood is transformed into the product of the BPR score between user and items proxy and the similarity between items proxy and keyphrases. The optimization objective is then converted to a lower bound, following Jensen's inequality we have:

$$\begin{aligned} &\log p\left(\mathcal{D}_C | U^*, V, K\right) \\ &= \sum_{\langle u_i, k_j \rangle \in \mathcal{D}_C} \log \sum_{v_s \in V} p\left(\langle u_i, v_s \rangle\right) \cdot p\left(v_s | k_j\right) \\ &\geqslant \sum_{\langle u_i, k_j \rangle \in \mathcal{D}_C} \mathbb{E}_{v_s \sim p(v_s | k_j)} \log p\left(\langle u_i, v_s \rangle\right) \end{aligned} \tag{8}$$

Therefore, the BPR score of users for critiquing keyphrases is cleverly approximated from the preference expectations of users for proxy items.

Through Eq.8, we can calculate the BPR score of user-keyphrases. Practically, in the procedure of constantly upgrading the recommendations, the user will make new critiquing in each cycle, which will be considered for updating the user embedding. The observed performance degradation of multi-step critiquing is probable due to the unavoidable loss of essential features during continuous fine-tuning of user embedding, which has been sufficiently examined in the field of continuous learning. Considering the demand for real-time response in critiquing, the methods of replay not only require additional computational time expense and storage space but also have the possibility of causing overfitting situations. Accordingly, preserving the crucial information of the original model by regularization methods while optimizing the user-keyphrase BPR score is more compatible with the requirements of critiquing. In addition to optimizing the user-keyphrases BPR score in Eq.5, we hope that the updated user embedding can retain the critical information from the prior.

Ultimately, the model will compute the BPR score between the user embedding posterior and the candidate items embedding and re-rank them to update the Top-K recommended items for users.

3.4. Training Strategy

The previous section introduced the philosophy of the IPGC framework. In this section, we will describe the specific training strategy.

Target Representative Items selection. In Eq.8 we calculate the user-keyphrases BPR scores indirectly through the items proxy, making it available for the backpropagation algorithm. A crucial aspect here is to figure out the association probability between the items proxy and keyphrase $p\left(v_s | k_j\right)$. Since the original model has learned rich relationships between items and keyphrases in KG, this value can be given either by similarity estimation or algorithms learning KG structural features in the original model [10, 13, 30]. However, because of their individual ways of learning KG representations, it is impracticable to generalize them.

Therefore, to commonly address Eq.8, we instead decided to utilize the basic structure of KG to represent the associations between items and keyphrases. Specifically, if a connection trail between them is observed in KG, we set $p\left(v_s | k_j\right)$ to 1, otherwise 0. Considering that sampling all items related to keyphrase in KG incurs excessive cost, we use the Monte Carlo method [22] for unbiased sampling approximations:

$$\begin{aligned} &\mathbb{E}_{v_s \sim p(v_s | k_j)} \log p\left(\langle u_i, v_s \rangle\right) \\ &\approx \frac{1}{M} \sum_{s=1}^{M} \log p\left(\langle u_i, v_s^* \rangle\right), v_s^* \sim p\left(v_s | k_j\right) \end{aligned} \tag{9}$$

where M is the number of samples. Since the learned KG representations in the original model encompass some higher-order information, to minimize the loss of this information due to sampling items proxy, it is necessary to sample higher-order items associated with keyphrases when using the Monte Carlo method. We adopt a multi-hop sampling method to extract l hop items related to keyphrase k from the knowledge graph:

$$\mathcal{V}^* = MultihopSample\left(l, r, k, \mathcal{G}\right) \tag{10}$$

where we sample items at different hops proportionally, and r denotes the ratio of items sampled from the l-th hop over

the total M sampled items. And we apply a simple random sampling strategy for items at each hop. Therefore, Eq.9 can be transformed into:

$$\begin{aligned}&\mathbb{E}_{v_s\sim p(v_s|k_j)}\log p\left(\langle u_i, v_s\rangle\right)\\&\approx\frac{1}{M}\sum_{s=1}^{M}\log p\left(\left\langle u_i, v_s^*\right\rangle\right), v_s^*\in\mathcal{V}^*\end{aligned}\tag{11}$$

Thus, we can tactfully substitute the BPR score of user for items proxy with the BPR score of the user against the keyphrase:

$$\log p\left(\left\langle u_i, v_s^*\right\rangle\right)=\log\sigma\left(h-\mathbf{u}_i{}^\top\mathbf{v}_s^*\right)\tag{12}$$

where h is a constant and σ denotes the sigmoid function. And the critiquing likelihood can be expressed as:

$$\begin{aligned}&\log p\left(\mathcal{D}_c|U^*,V,K\right)\\&=\eta\cdot\frac{1}{M}\sum_{\langle u_i,k_j\rangle\in\mathcal{D}_c}\sum_{s=1}^{M}\log\sigma\left(h-\mathbf{u}_i{}^\top\mathbf{v}_s^*\right)\end{aligned}\tag{13}$$

where η=1 means k is positive critiquing and η=-1 represents negative critiquing. Alternatively, IPGC provides the option of involving (u, i) pairs from $\mathcal{D}_T$ in updating the user embedding when user critiquing is negative:

$$\begin{aligned}&\log p\left(\mathcal{D}_c|U^*,V,K\right)\\&=-\frac{1}{M}\sum_{\langle u_i,k_j\rangle\in\mathcal{D}_c,\langle u_i,v^+\rangle\in\mathcal{D}_T}\sum_{s=1}^{M}\log\sigma\left(\mathbf{u}_i{}^\top\mathbf{v}^+-\mathbf{u}_i{}^\top\mathbf{v}_s^*\right)\end{aligned}\tag{14}$$

Continuous Critiquing. In section 3.3 we analyzed the problem of recommendation performance collapse in continuous critiquing and raised the prospect of alleviating it by inhibiting key feature change within user embedding prior to using regularization. At this point, our goal is to mitigate catastrophic forgetting by tokenizing critical parameters and penalizing the tendency towards modifying those parameters. We are inspired by previous approaches [1], which estimate these emphasis weights approximate the sensitivity of the learned function in the original model. Notice that the method was designed for computer vision and requires adaptation for critiquing tasks; we focus only on the last layer of the original recommendation model which is to compute the feature-crossing of any candidate item v for each user u. Since the item embedding is fixed invariant, we only care about the sensitivity of the user embedding against perturbation. For a given (u, v) pair, we use $\phi\left(u_v;\theta_u\right)$ to denote its BPR score and $\theta_u=\left\{u^{(i)}\right\}$ to denote the user embedding tensor. Then the perturbation $\delta=\left\{\delta_i\right\}$ to θ_u resulting in change in the final predicted value can be expressed as:

$$\phi\left(u_v;\theta_u+\delta\right)-\phi\left(u_v;\theta_u\right)\approx\sum_i g_i\left(u_v\right)\delta_i\tag{15}$$

Where the value of $g_i(u)=\frac{\partial\|\phi(u_v;\theta_u)\|}{\partial\theta_u^{(i)}}$ can measure the significance of the weight, indicating how much the current prediction will change with a perturbation to this parameter. Finally, we can obtain the importance weight of each user:

$$\Omega_u=\frac{1}{N}\sum_{v=1}^{N}\left\|g_i\left(u_v\right)\right\|\tag{16}$$

where N is the total interaction number of users in the training set. Modifications to parameters with minor importance weights will not have a significant impact on the output, and they will not be overly constrained in further tasks. However, changes to parameters with higher importance weights will be restricted or preferably left unchanged, we have:

$$\log p\left(U^*\right)=\sum\Omega_U\left\|U-U^*\right\|\tag{17}$$

Update embedding. Finally, the maximum posterior probability of the user embedding can be transformed into minimizing the loss function. Therefore, for the proposed IPGC framework, we have the following loss function:

$$\begin{aligned}\min\mathcal{L}_{IPGC}&=-\log\left(p\left(\mathcal{D}_c|U^*,V,K\right)\cdot p\left(U^*\right)\right)\\&=-\eta\cdot\frac{1}{M}\sum_{\langle u_i,k_j\rangle\in\mathcal{D}_c}\sum_{s=1}^{M}\log\sigma\left(h-\mathbf{u}_i{}^\top\mathbf{v}_s^*\right)\\&\quad-\lambda_\Omega\sum\Omega_U\left\|U-U^*\right\|\end{aligned}\tag{18}$$

Where λ_Ω is a hyperparameter and $v_s^*\in\mathcal{V}^*$.

4. EXPERIMENTS

We present empirical results to substantiate the efficacy of our proposed IPGC. The experiment is intended to answer the following research questions:

RQ1: Does the proposed IPGC effectively refine the recommendation in terms of critiquing information? How does the result compare with the current state-of-the-art critiquing methods regarding generalizability and performance?

RQ2: Does the knowledge graph manifest a positive influence on finding items proxy, and does the anti-forgetting regularizer actually work?

RQ3: Different hyperparameter settings (e.g., number of multi-hop sampling layers, effect of total number of samples, etc.) and case study.

4.1. Dataset Description

We validate IPGC on two real-world datasets for movie and music in the experiments: (1) We use the Movielens-1M, a widely used movie benchmark dataset released by RippleNet [32]; and (2) Last-FM, a music benchmark dataset collected from the Last.fm online music system released by KGAT [38]. For Movielen-1M, since it incorporates the user's explicit scores of the movie (rating from 1 to 5), we convert it to implicit feedback (with setting the rating threshold to 1). As for Last-FM we follow exactly the protocol

Table 1
Statistics of MovieLens-1M and Last-FM

		Movielens-1M	Last-FM
User-Item Interaction	Users	6036	23566
	Items	2445	48123
	Interactions	376886	1712638
Knowledge Graph	Entities	182011	106389
	Keyphrases	179566	58266
	Triplets	309172	464567

given by KGAT for processing and slicing the dataset in order to avoid inaccuracy. The statistical information of these datasets is summarized in Table 1. Keyphrases refer to the entities in the knowledge graph that are not items. We split the datasets into training and testing sets at 8:2. The other training settings for KGAT, KGIN and DiffKG followed the configurations provided in their papers. We stopped training the original model until it got the highest recommendation score. In the evaluation phase, we use the all-ranking strategy to evaluate the recommendations, which is to rank all the items excluding the training set. Our codes are shared on https://github.com/StZHY/Critique/tree/master.

4.2. Experiment Settings

We compare IPGC with CE-VAE [18], BK-VAE [44], DCE-VAE [23] and BCIE [29] on two real-world datasets: the MovieLens-1M and Last-FM. During the inference phase, adopting the familiar settings from previous studies [44, 23], we conducted a total of 10 rounds of multistep critiquing experiments. In order to simulate real user feedback without exposing the practical test dataset, we used only negative critiquing (positive critiquing will inevitably include attributes of the items in the test set). By using a program independent of IPGC, we calculated the differences between the average frequency of keyphrases occurrences corresponding to the Top-20 items predicted by the model in each round and corresponding to the target items (the results of the first round were taken from the predictions of the original model). We then selected the top-5 keyphrases with the largest differences as the user critiquing. To evaluate the effectiveness of the critiquing model, we used three metrics, *Recall@5*, *NDCG@5*, and *HR*@5, to measure the effectiveness in refining recommendations based on multi-step critiquing. It should be emphasized that, as mentioned in previous studies [44, 23], meaningful comparisons of performance trends can only be conducted if the starting scores are roughly the same. However, in previous work, as shown in Table 2, the base recommendation is significantly less than the KG-based approach, and the VAE model is not suitable for general KG recommendation task. To ensure fair comparisons, we have re-implemented their methodology in the KG recommender system, although that may cause potential issues. Our codes are shared on https://github.com/StZHY/IPGC/tree/master.

4.3. Baselines

We introduce three state-of-the-art KG recommender models **KGAT** [38], **KGIN** [39] and **DiffKG** [11] and implement IPGC on top of them. Next, we present baseline critiquing methods and detail how we apply these methods towards KG recommender systems:

CE-VAE [18] operates by directly setting the weights of the VAE latent variable corresponding to critiquing keyphrase embeddings to zero. In the KG recommender framework, we exclude critiquing keyphrase from user embeddings aggregation.

BK-VAE [44] uses a Bayesian algorithm to update the user embedding through modify the corresponding keyphrases of users. In this paper, we update the user embeddings by directly constructing a BPR loss function with the user embeddings and keyphrase embeddings.

DCE-VAE [23] further clarified critiquing keyphrases based on BK-VAE by using a keyphrases tree to capture more precise meanings of user responses. Since KG itself consists of a large amount of refined knowledge, we build on BK-VAE by mapping the keyphrases to all direct neighboring nodes in the KG to refine the meaning of the keyphrases.

BCIE [29]: It models triplets (user, like, items) in the knowledge graph through knowledge representation methods and transmits user feedback using belief propagation methods. We traverse the set of all nodes in the KG on the relation paths between the user's historical items and the critiquing keyphrase nodes as the object to update recommendations.

4.4. Parameter Settings

We implement the IPGC in PyTorch. For a fair comparison, we set the optimization method as Adam [12], and the learning rate is to 0.005, specially fixing this value for all datasets and comparison methods. We use the grid search to explore the parameters of the IPGC. First the regularization parameter is set λ_Ω between $\{10^{-2}, 10^{-3}, 10^{-4}\}$, the hop number for sampling is 2, and the probability r of sampling from l=1 is tuned in $\{1.0, 0.8, 0.5, 0.0\}$.For the total number M of items proxy sampled for each keyphrase k is chosen from $\{1, 5, 10\}$. And for the two optimization for *IPGC*, where the one using (u, i) pairs in the training set is named *IPGC-t*.

4.5. Performance Comparison (RQ1)

The results of the multi-step critique are reported in Figure 3, 4 and 5, where the trend of the scores signifies the ability of each method to refine recommendations.

- DiffKG, KGIN and KGAT demonstrate better performance on three VAE-based metrics and BCIE in Table 2. A better original model can serve a superior user experience, while increasing performance on the better model is usually more challenging.

- IPGC exhibits significantly competitive performance compared to other critiquing methods. In a continuous 10-step critiquing on e.g., Last-FM, IPGC can raise the

Table 2
Performance of different original methods.

	MovieLens-1M			Last-FM		
	Recall@5	NDCG@5	HR@5	Recall@5	NDCG@5	HR@5
CE-VAE	0.0265	0.0680	0.1701	0.0096	0.0191	0.0379
BK-VAE	0.0324	0.0793	0.2037	0.0115	0.0232	0.0462
DCE-VAE	0.0370	0.0842	0.2171	0.0117	0.0231	0.0463
BCIE	0.0272	0.0690	0.1559	0.0238	0.0397	0.0745
KGAT	0.0897	0.1967	0.5172	0.0399	0.0612	0.1789
KGIN	0.1327	0.2687	0.6368	0.0483	0.0743	0.1960
DiffKG	0.1275	0.2589	0.6322	0.0529	0.0843	0.2154

original scores of KGIN by 54%, 36%, and 40% for the metrics *NDCG@5*, *Recall@5*, and *HR@5*, respectively. Compared with the Sota critiquing methods, it has significant improvement. Although our simulation experiments represent an ideal situation, such a dramatic benefit is a validation of IPGC's excellent capability to refine recommendations in different methods and datasets.

- The improvement on KGAT, KGIN and DiffKG is approximately similar, indicating that the selection of the original model does not impact IPGC. This suggests our model, as a plug-in, can be adapted to all general KG recommender models and can produce good results.
- Compared to IPGC, IPGC-t reinforces user preferences using examples from the training data and performs better in KGAT and DiffKG. We conclude this result may be related to the initial model design.
- CE-VAE, BK-VAE, DCE-VAE and BCIE perform not well. This is because they attempt to directly optimize user-keyphrase preference scores, which is effective for specially designed VAE models but not for a general KG-based model. Some methods do not even have any proactive effect. Furthermore, all these methods exhibit the typical characteristics of catastrophic forgetting in multi-step critiquing task, indicating that the problem we have raised is not an illusory one. Moreover, the weaker performance of these methods implicitly demonstrates the validity of mining items proxy to indirectly compute the BPR score of users for keyphrases.

4.6. Performance of Positive Critiquing (RQ1)

Despite what we assume in the real world, most critiquing is negative because when users are clearly expressive of their preferences, they usually know what they want and rarely rely on recommender systems to shop for their favorite items. However, especially in conversational recommendation systems, a high probability of positive critiquing arises. We also conducted an exploration of positive critiquing, even though there will be a possibility that the test data will be leaked to the critiquing model. As illustrated in Figure 6, we observed that even with positive critiquing, IPGC still exhibits better performance and the trend does not diminish with increasing critique rounds. Noteworthy that even with positive feedback, baseline models still suffer catastrophic forgetting, indicating that IPGC constructs user-keyphrase correlation through items proxy is more reasonable.

4.7. Analysis of the Items Proxy (RQ2)

The effectiveness of items proxy has been demonstrated, but it is worth exploring whether the knowledge graph plays a key role in items proxy selection. We use Table 3 to visually illustrate the comparison between target items proxy from the KG and using random sampling. And it is obvious from the results that KG takes a crucial part in the search of items proxy. Besides, random sampling helps to improve recommendations is evident in the Movielens dataset. This is because during the sampling process, there is always a chance of picking items that are relevant to user critiquing, leading to some improvements. However, in the sparser Last-FM dataset, random sampling did not produce any significant effect.

4.8. Efficacy of Anti-forgotten Regularizer (RQ2)

In Section 4.5, most other methods suffer from catastrophic forgetting after 2 or 3 rounds. In contrast, both IPGC and IPGC-t manage to avoid the inference of forgetting and show stable tuning capabilities. However, the stable refinement of IPGC and IPGC-t could also due to the fact that the learning rate is set too small. To eliminate the effect of this factor, we consider exploring the utility of the anti-forgetting regularizer. in Figure 7(c), where $\lambda_\Omega = 0$ represents the case without the regularizer, it is obvious that the regularizer does provide a great deal of assistance in preventing catastrophic forgetting. Without it, the model will struggle to growth after steps with the recommendation declining. its overall performance is weaker compared to IPGC with the regularizer. That suggests that regularizer are effective in inhibiting the variation of key weights and successfully preserving the user's initial preferences.

4.9. Hyperparametric Analysis (RQ3)

In this section we will discuss some of the hyperparameters we set in the IPGC.

4.9.1. Impact of M

The quality of sampling representative items proxy for keyphrases is crucial. The goal is to explore an adequate sampling size nearest to conveying the semantics

Table 3
Comparison of sampling according to KG and random sampling. The metrics are Recall@5, NDCG@5 ,and HR@5.

Steps	IPGC(KGIN)						IPGC(KGIN)-random sampling					
	Last-FM			Movielens-1m			Last-FM			Movielens-1m		
	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG	HR
0	0.0483	0.0743	0.196	0.1327	0.2687	0.6368	0.0483	0.0743	0.196	0.1327	0.2687	0.6368
1	0.0517	0.0814	0.2124	0.1367	0.2804	0.6438	0.0483	0.0742	0.1957	0.1323	0.2694	0.6368
3	0.0584	0.0946	0.2445	0.1459	0.3036	0.6668	0.0479	0.0735	0.1945	0.1334	0.2716	0.6375
6	0.0644	0.1076	0.2682	0.1535	0.3202	0.6803	0.0474	0.0726	0.1913	0.1357	0.275	0.6412
10	0.0655	0.1145	0.275	0.1554	0.3289	0.681	0.0467	0.0717	0.1909	0.1345	0.2752	0.6381
MaxImp	36%	54%	40%	17%	22%	6%	0%	0%	0%	2%	2%	0.60%

of keyphrase. We conducted an exploration of sampling quantities in $M = \{1, 5, 10\}$. As depicted in Figure 7(a), a few samples ($M = 1$) cannot fully capture the characteristics of keyphrases. Conversely, too many samples ($M = 10$) will result in a downward trend toward the tail of multistep critiquing. Choosing $M = 5$ was based on the fact that despite a slightly weaker performance in the first few rounds, it maintained its ability well all over.

4.9.2. Impact of r

In this section, we explore the impact of higher-order items on surrogate keyphrases. Since the original model utilized considerable higher-order information during training phrase to enhance the keyphrase embedding, it is worthwhile to investigate how many higher-order items need to be sampled. We selected the proportion of one-hop items directly connected with keyphrases in $r = \{1.0, 0.8, 0.5, 0.0\}$. As illustrated in Figure 7(b), extracting more higher-order items in the preliminary usually yields greater improvements. However, later on, fatigue will occur, causing decreased benefits.

4.9.3. Impact of λ_Ω

The strength of the regularizer has a great influence on the resistance of IPGC against forgetting. In Figure 7(c), while higher λ_Ω make the curve very smooth, but also greatly reduces its ability. And if λ_Ω is too small there will be some volatility in the critiquing process. Therefore, we operate for $\lambda_\Omega = 0.001$, ensuring that the regularizer effective without compromising IPGC's adaptability.

4.10. Case Study (RQ3)

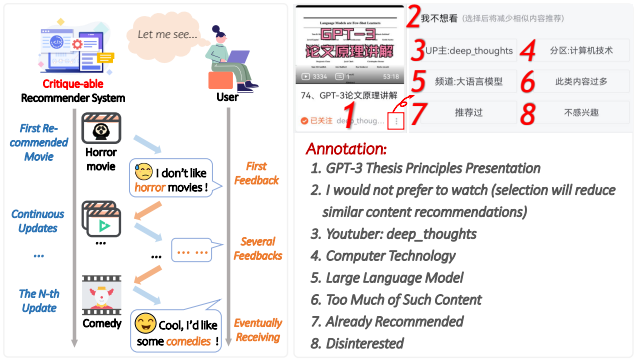
We illustrate the critiquing using a real case from the Movielens-1M. We randomly selected a user with his ID is 64 and demonstrated the simulation of his critiquing process. As seen in Figure 8, he disliked the 2nd and 4th movies in the original recommendation. Therefore, based on the explanatory keyphrases provided by the application, he made four valid negative critiquing: Spanish, Drama, Westerns, and Phillip Noyce. Based on these, IPGC update Top 5 recommendations with two films that users genuinely enjoyed while retaining the previous hits. This example illustrates that our IPGC framework can precisely refine recommendations according to user critiquing on the basis of the original model that is equipped with a high degree of accuracy.

5. Conclusion and Future Work

In this work, we introduced IPGC, a novel and versatile plugin designed to work with mostly CF models upon KG, providing a new option for refining recommendations using critiquing in list-based recommender scenarios. Experimental results demonstrate the effectiveness of the IPGC under various types of critiquing. It can be implemented in different architectures of KG recommender models and effectively alleviates catastrophic forgetting. Although IPGC still has some shortcomings, such as the necessity to work dependent on KG and the anti-forgotten regularization is not sota, the items proxy mechanism can efficiently utilize critiquing information to refine recommendations. For future work, we plan to (1) develop a more sophisticated proxy selection mechanism and (2) design a more powerful regularizer to enhance the performance of the critiquing plugin.

Acknowledgements

This work is supported by National Natural Science Foundation of China under 62277028.



(a) Continuous critiquing **(b) A real critiquing interface**

Figure 1: (a) Illustration of continuous critiquing interactions (b) Alternative critiquing keyphrases available from the video app.

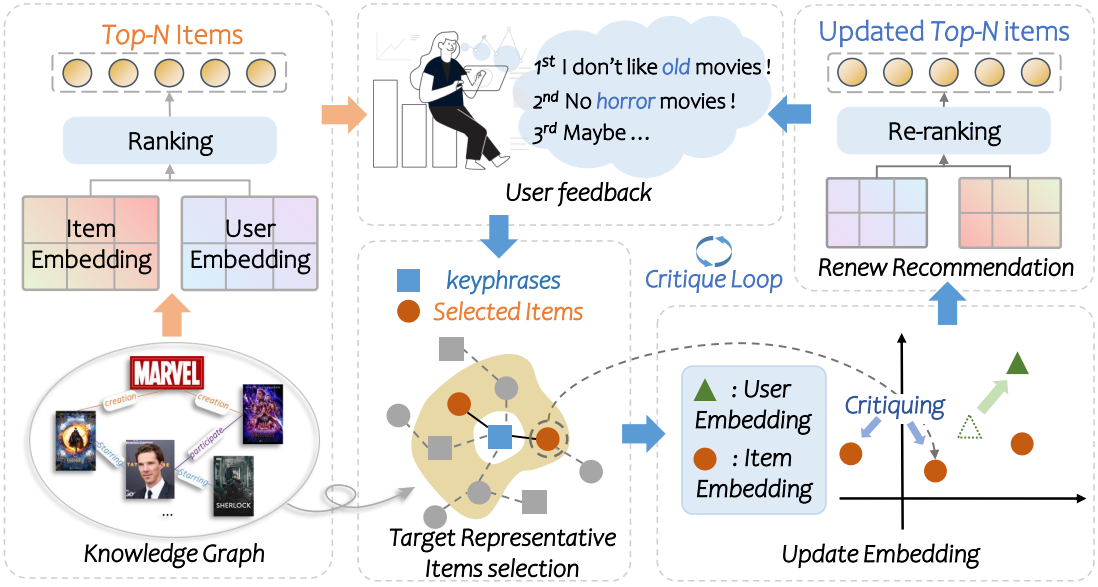


Figure 2: Illustration of the proposed IPGC, which presents the whole procedure of how the framework functions in the KG recommender system to allow critiquing for users.

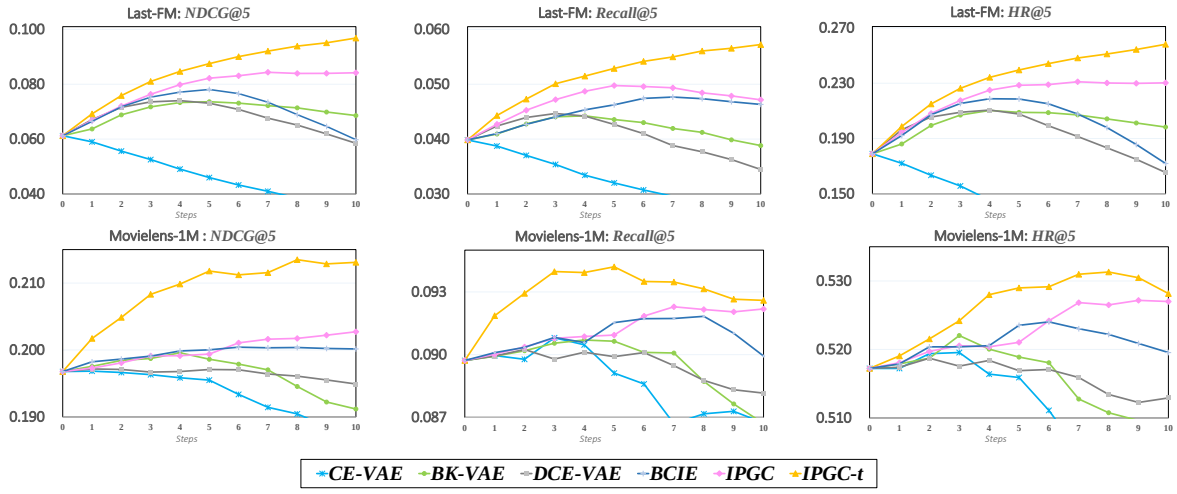


Figure 3: Comparison of IPGC and baseline methods on *KGAT* for *NDCG@K*, *Recall@K* and *HR@K*, with dataset MovieLens and Last-FM.

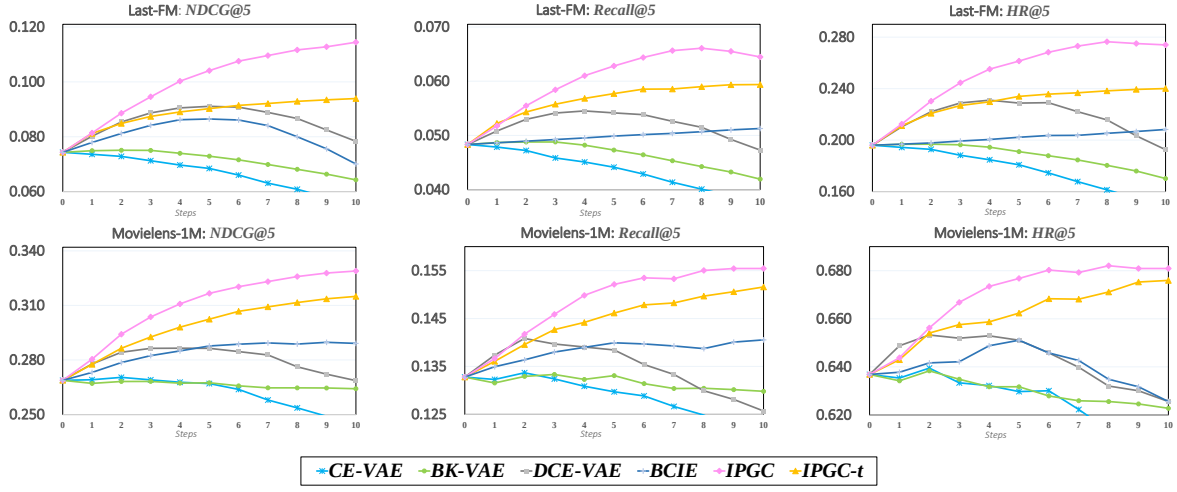


Figure 4: Comparison of IPGC and baseline methods on *KGIN* for *NDCG@K*, *Recall@K* and *HR@K*, with dataset MovieLens and Last-FM.

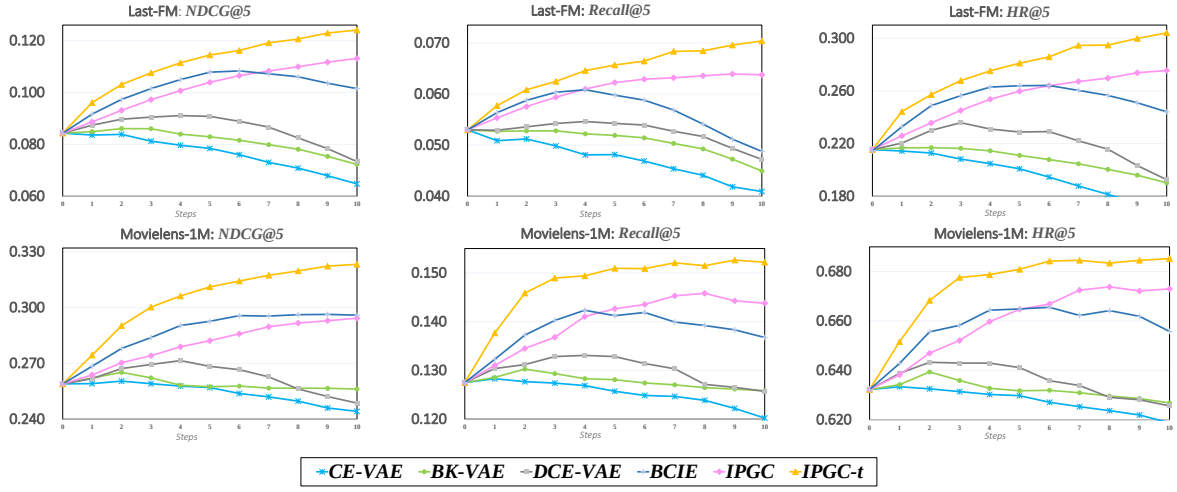


Figure 5: Comparison of IPGC and baseline methods on *DiffKG* for *NDCG@K*, *Recall@K* and *HR@K*, with dataset MovieLens and Last-FM.

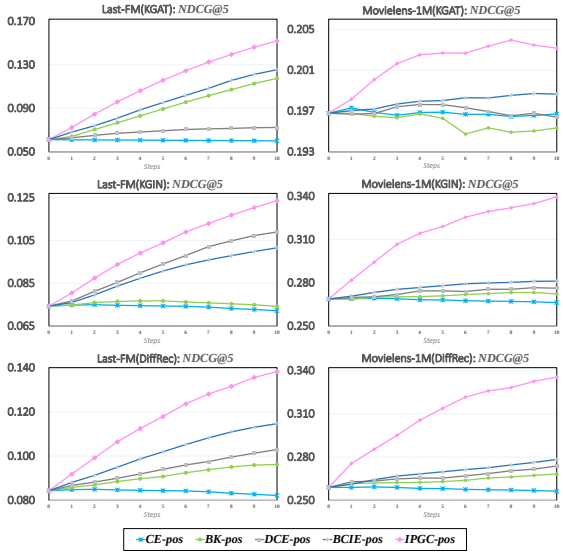
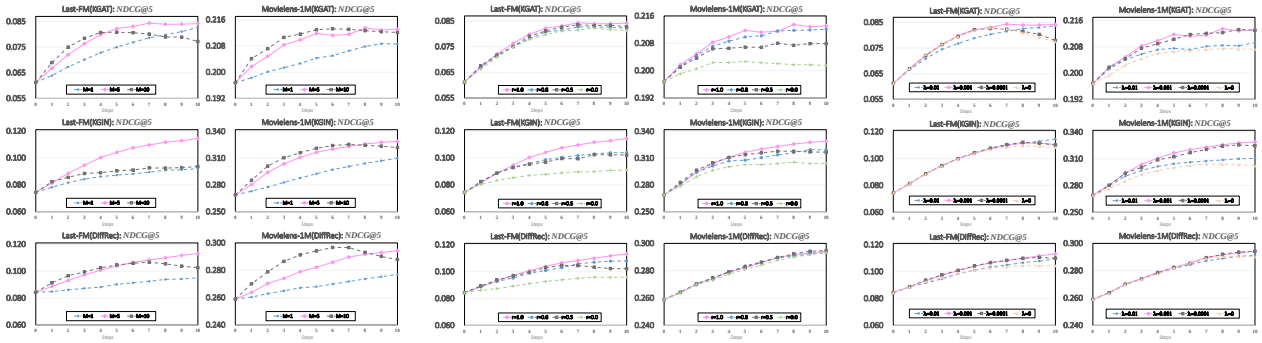


Figure 6: Comparison of IPGC and baselines based on KGAT, KGIN and DiffKG using positive critiquing at NDCG@5.



(a) Impact of hyperparametric M. (b) Impact of hyperparametric r. (c) Impact of hyperparametric λ_Ω.

Figure 7: Influence of different hyperparameters

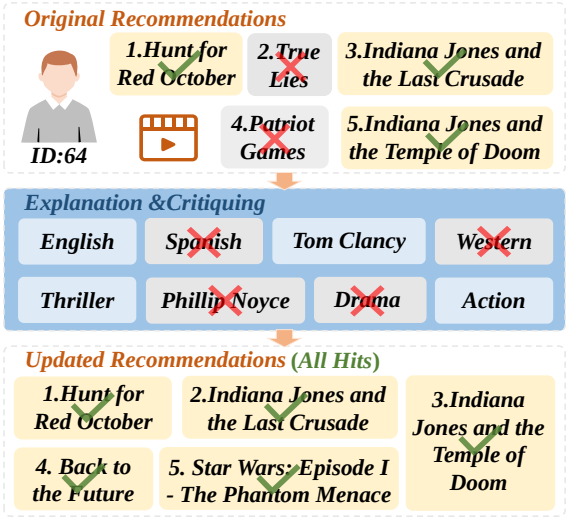


Figure 8: Example of a user ID 64 in Movielens.

References

[1] Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T., 2018. Memory aware synapses: Learning what (not) to forget, in: Proceedings of the European conference on computer vision (ECCV), pp. 139–154.

[2] Antognini, D., Faltings, B., 2021. Fast multi-step critiquing for vae-based recommender systems, in: Proceedings of the 15th ACM Conference on Recommender Systems, pp. 209–219.

[3] Antognini, D., Faltings, B., 2022. Positive and negative critiquing for vae-based recommenders. arXiv preprint arXiv:2204.02162 .

[4] Antognini, D., Musat, C., Faltings, B., 2020. Interacting with explanations through critiquing. arXiv preprint arXiv:2005.11067 .

[5] Dai, Q., Wu, X.M., Fan, L., Li, Q., Liu, H., Zhang, X., Wang, D., Lin, G., Yang, K., 2022. Personalized knowledge-aware recommendation with collaborative and attentive graph convolutional networks. Pattern Recognition 128, 108628.

[6] Friedman, L., Ahuja, S., Allen, D., Tan, T., Sidahmed, H., Long, C., Xie, J., Schubiner, G., Patel, A., Lara, H., et al., 2023. Leveraging large language models in conversational recommender systems. arXiv preprint arXiv:2305.07961 .

[7] Gong, J., Zhao, Y., Zhao, J., Zhang, J., Ma, G., Zheng, S., Du, S., Tang, J., 2024. Personalized recommendation via inductive spatiotemporal graph neural network. Pattern Recognition 145, 109884.

[8] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S., 2017. Neural collaborative filtering, in: Proceedings of the 26th international conference on world wide web, pp. 173–182.

[9] Huang, X., Lian, J., Lei, Y., Yao, J., Lian, D., Xie, X., 2023. Recommender ai agent: Integrating large language models for interactive recommendations. arXiv preprint arXiv:2308.16505 .

[10] Ji, G., He, S., Xu, L., Liu, K., Zhao, J., 2015. Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers), pp. 687–696.

[11] Jiang, Y., Yang, Y., Xia, L., Huang, C., 2024. Diffkg: Knowledge graph diffusion model for recommendation, in: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, pp. 313–321.

[12] Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

[13] Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 .

[14] Li, H., Sanner, S., Luo, K., Wu, G., 2020. A ranking optimization approach to latent linear critiquing for conversational recommender systems, in: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 13–22.

[15] Li, Q., Ma, H., Zhang, R., Jin, W., Li, Z., 2023. Dual-view co-contrastive learning for multi-behavior recommendation. Applied Intelligence , 1–18.

[16] Li, X., She, J., 2017. Collaborative variational autoencoder for recommender systems, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 305–314.

[17] Luo, K., Sanner, S., Wu, G., Li, H., Yang, H., 2020a. Latent linear critiquing for conversational recommender systems, in: Proceedings of The Web Conference 2020, pp. 2535–2541.

[18] Luo, K., Yang, H., Wu, G., Sanner, S., 2020b. Deep critiquing for vae-based recommender systems, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, pp. 1269–1278.

[19] Ma, W., Zhang, M., Cao, Y., Jin, W., Wang, C., Liu, Y., Ma, S., Ren, X., 2019. Jointly learning explainable rules for recommendation with knowledge graph, in: The world wide web conference, pp. 1210–1221.

[20] Nema, P., Karatzoglou, A., Radlinski, F., 2021. Disentangling preference representations for recommendation critiquing with ß-vae, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 1356–1365.

[21] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L., 2012. Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 .

[22] Shapiro, A., 2003. Monte carlo sampling methods. Handbooks in operations research and management science 10, 353–425.

[23] Shen, T., Mai, Z., Wu, G., Sanner, S., 2022. Distributional contrastive embedding for clarification-based conversational critiquing, in: Proceedings of the ACM Web Conference 2022, pp. 2422–2432.

[24] Shen, X., Yi, B., Liu, H., Zhang, W., Zhang, Z., Liu, S., Xiong, N., 2019. Deep variational matrix factorization with knowledge embedding for recommendation system. IEEE Transactions on Knowledge and Data Engineering 33, 1906–1918.

[25] Shenbin, I., Alekseev, A., Tutubalina, E., Malykh, V., Nikolenko, S.I., 2020. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback, in: Proceedings of the 13th international conference on web search and data mining, pp. 528–536.

[26] Shu, Y., Gu, H., Zhang, P., Zhang, H., Lu, T., Li, D., Gu, N., 2023. Rah! recsys-assistant-human: A human-central recommendation framework with large language models. arXiv preprint arXiv:2308.09904 .

[27] Sun, R., Cao, X., Zhao, Y., Wan, J., Zhou, K., Zhang, F., Wang, Z., Zheng, K., 2020. Multi-modal knowledge graphs for recommender systems, in: Proceedings of the 29th ACM international conference on information & knowledge management, pp. 1405–1414.

[28] Tang, Z., Floto, G., Toroghi, A., Pei, S., Zhang, X., Sanner, S., 2023. Logicrec: Recommendation with users' logical requirements. arXiv preprint arXiv:2304.11722 .

[29] Toroghi, A., Floto, G., Tang, Z., Sanner, S., 2023. Bayesian knowledge-driven critiquing with indirect evidence. arXiv preprint arXiv:2306.05636 .

[30] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903 .

[31] Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., Liu, Q., 2018a. Shine: Signed heterogeneous information network embedding for sentiment link prediction, in: Proceedings of the eleventh ACM international conference on web search and data mining, pp. 592–600.

[32] Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., Guo, M., 2018b. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: Proceedings of the 27th ACM international conference on information and knowledge management, pp. 417–426.

[33] Wang, H., Zhang, F., Xie, X., Guo, M., 2018c. Dkn: Deep knowledge-aware network for news recommendation, in: Proceedings of the 2018 world wide web conference, pp. 1835–1844.

[34] Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W., Wang, Z., 2019a. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 968–977.

[35] Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., Guo, M., 2019b. Multi-task feature learning for knowledge graph enhanced recommendation, in: The world wide web conference, pp. 2000–2010.

[36] Wang, H., Zhao, M., Xie, X., Li, W., Guo, M., 2019c. Knowledge graph convolutional networks for recommender systems, in: The world wide web conference, pp. 3307–3313.

[37] Wang, W., Feng, F., Nie, L., Chua, T.S., 2022a. User-controllable recommendation against filter bubbles, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1251–1261.

[38] Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S., 2019d. Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 950–958.

[39] Wang, X., Huang, T., Wang, D., Yuan, Y., Liu, Z., He, X., Chua, T.S., 2021. Learning intents behind interactions with knowledge graph for recommendation, in: Proceedings of the web conference 2021, pp. 878–887.

[40] Wang, X., Liu, K., Wang, D., Wu, L., Fu, Y., Xie, X., 2022b. Multi-level recommendation reasoning over knowledge graphs with reinforcement learning, in: Proceedings of the ACM Web Conference 2022, pp. 2098–2108.

[41] Wang, Z., Lin, G., Tan, H., Chen, Q., Liu, X., 2020. Ckan: Collaborative knowledge-aware attentive network for recommender systems, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, pp. 219–228.

[42] Wu, G., Luo, K., Sanner, S., Soh, H., 2019. Deep language-based critiquing for recommender systems, in: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 137–145.

[43] Xia, L., Huang, C., Shi, J., Xu, Y., 2023. Graph-less collaborative filtering, in: Proceedings of the ACM Web Conference 2023, pp. 17–27.

[44] Yang, H., Shen, T., Sanner, S., 2021. Bayesian critiquing with keyphrase activation vectors for vae-based recommender systems, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2111–2115.

[45] Yang, Y., Huang, C., Xia, L., Huang, C., 2023. Knowledge graph self-supervised rationalization for recommendation, in: Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining, pp. 3046–3056.

[46] Yang, Y., Huang, C., Xia, L., Li, C., 2022. Knowledge graph contrastive learning for recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1434–1443.

[47] Zenke, F., Poole, B., Ganguli, S., 2017. Continual learning through synaptic intelligence, in: International conference on machine learning, PMLR. pp. 3987–3995.

[48] Zhai, Y., Tong, S., Li, X., Cai, M., Qu, Q., Lee, Y.J., Ma, Y., 2023. Investigating the catastrophic forgetting in multimodal large language models. arXiv preprint arXiv:2309.10313 .

[49] Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y., 2016. Collaborative knowledge base embedding for recommender systems, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 353–362.

[50] Zhang, G., Li, D., Gu, H., Lu, T., Shang, L., Gu, N., 2023a. Simulating news recommendation ecosystem for fun and profit. arXiv preprint arXiv:2305.14103 .

[51] Zhang, H., Shen, X., Yi, B., Wang, W., Feng, Y., 2023b. Kgan: Knowledge grouping aggregation network for course recommendation in moocs. Expert Systems with Applications 211, 118344.

[52] Zhang, X., Xin, X., Li, D., Liu, W., Ren, P., Chen, Z., Ma, J., Ren, Z., 2023c. Variational reasoning over incomplete knowledge graphs for conversational recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pp. 231–239.

[53] Zhou, K., Zhao, W.X., Bian, S., Zhou, Y., Wen, J.R., Yu, J., 2020. Improving conversational recommender systems via knowledge graph based semantic fusion, in: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 1006–1014.

[54] Zou, D., Wei, W., Mao, X.L., Wang, Z., Qiu, M., Zhu, F., Cao, X., 2022a. Multi-level cross-view contrastive learning for knowledge-aware recommender system, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1358–1368.

[55] Zou, D., Wei, W., Wang, Z., Mao, X.L., Zhu, F., Fang, R., Chen, D., 2022b. Improving knowledge-aware recommendation with multi-level interactive contrastive learning, in: Proceedings of the 31st ACM international conference on information & knowledge management, pp. 2817–2826.