A Plug-in Critiquing Approach for Knowledge Graph Recommendation Systems via Representative Sampling.

Anonymous Author(s)

ABSTRACT

Incorporating a critiquing component into recommender applications facilitates the enhancement of user perception. Typically, critique-able recommender systems adapt the model parameters and update the recommendation list in real-time through the analysis of user critiquing keyphrases in the inference phase. The current critiquing methods necessitate the designation of a dedicated recommendation model to estimate user relevance to the critiquing keyphrase during the training phase preceding the recommendations update. This paradigm restricts the applicable scenarios and reduces the potential for keyphrase exploitation. Furthermore, these approaches ignore the issue of catastrophic forgetting caused by continuous modification of model parameters in multi-step critiquing. Thus, we present a general ***R**epresentative **I**tems **S**ampling Framework for **C**ritiquing on Knowledge Graph Recommendation* (RISC) implemented as a plug-in, which offers a new paradigm for critiquing in mainstream recommendation scenarios. RISC leverages the knowledge graph to sample important representative items as a hinge to expand and convey information from user critiquing, indirectly estimating the relevance of the user to the critiquing keyphrase. Consequently, the necessity for specialized user-keyphrase correlation modules is eliminated with respect to a variety of knowledge graph recommendation models. Moreover, we propose a ***W**eight **E**xperience **R**eplay* (WER) approach based on KG to mitigate catastrophic forgetting by reinforcing the user's prior preferences during the inference phase. Our extensive experimental findings on three real-world datasets and three knowledge graph recommendation methods illustrate that RISC with WER can be effectively integrated into knowledge graph recommendation models to efficiently utilize user critiquing for refining recommendations and mitigate catastrophic forgetting. Our codes are shared on https://anonymous.4open.science/r/Critique-44F8.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Critiquing, Recommendation, Collaborative Filtering, Knowledge Graph

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Anonymous Author(s). 2018. A Plug-in Critiquing Approach for Knowledge Graph Recommendation Systems via Representative Sampling.. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Modern recommendation systems place a greater emphasis on the user experience while researching on enhancing recommendation accuracy[25, 36]. Numerous applications have devised channels for users to critique. Figure 1 illustrates screenshots of feedback interfaces for two different video software, wherein users can specify the keyphrases for why they dislike these videos. Afterward, these videos will be removed from the recommended list and substituted with new ones, significantly improving user satisfaction.

The recent research on critique-able recommender systems has concentrated on conversational recommendation systems (CRS)[17, 22, 28, 47]. These methods typically entail two steps. During the training phase, an original model is trained based on the user's historical data to generate initial recommendations. In the subsequent inference phase, the model analyzes the user's positive or negative critiquing on items or keyphrases presented in the communication dialogue, updates the model parameters and gradually refines the recommendations. Due to the explicit semantic information provided by keyphrases, the majority of research focuses more on keyphrases. Despite the demonstrated ability of these methodologies to adapt recommendations with critiquing in CRS, three concerns remain to be addressed. Firstly, the previous approaches have all designed specifically mechanisms that can directly model the correlation between users and keyphrases in the training phase for the benefit of comprehending critiquing in the inference phase. These strategies, while enabling the systems to handle critiquing, restrict the available scenarios and exhibit weak initial recommendation performance. Secondly, these methods generally neglect to consider collaborative critiquing information, lacking the adjustment of items with similar content and making relatively low utilization of keyphrases. Finally, these efforts have overlooked the cumulative change in successive parameter fine-tuning during the inference phase, which can reduce the capacity of the model to capture the user's old preferences, leading to catastrophic forgetting [4, 20, 35].

In order to leverage critiquing to more general recommendation scenarios in a reasonable manner, considering the knowledge graph (KG) enhanced recommendation models [10, 37, 38], which are powerful and contain rich knowledge that assists in efficiently understanding user feedback [18, 27, 39]. We prefer to implement critiquing based on knowledge graph recommender systems. Nevertheless, the attempt to generalize the critiquing framework based on knowledge graph recommendation models is inherently challenging. Most of these algorithms are formulated on collaborative

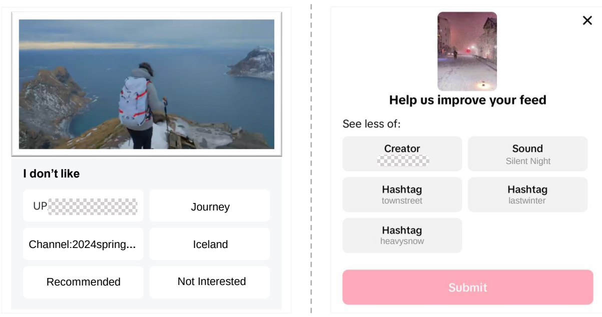

Figure 1: Illustration of critiquing interfaces from two video applications. Both software provide users with a wide range of alternative keyphrases available for critiquing as the reason for "not interested".

filtering (CF) strategies [8, 45], which only model the direct relevance of users to items and lack a straightforward correlation between users and keyphrases. Thus, it is necessary to develop a mechanism that can be universally applied to the KG recommendation model to estimate the relevance between users and keyphrases and to realize the mining of collaborative critiquing information. Moreover, the detrimental effects of catastrophic forgetting, which have been ignored in previous research, dictate that our critique-able model must possess the capability to preserve prior knowledge.

This paper introduces a generic Representative Items Sampling framework for Critiquing on Knowledge Graph Recommendation (RISC), which operates as a plug-in to mostly KG recommender systems. RISC presents a mechanism for sampling important representatives with the objective of estimating the relevance of the users to keyphrases. Given that the KG recommendation model normally aggregates the neighborhood information of items during the training phase to optimize parameters, we believe that selecting items as representatives to deliver information for critiquing keyphrases is a reasonable choice. The employment of representative items permits us to concentrate only on the direct relationships between users and items during the inference phase, aligning naturally with the original CF model. The optimal approach for ensuring the integrity of information delivery is to select the entire item neighbors as the representatives. This strategy minimizes the potential for loss and noise in the transmission of information. However, given the large number of neighbors, these nodes possess within the KG participating in the model training is impractical. Sampling important representative items as an estimate of the relevance of users to critiquing keyphrases can best accomplish our expectations. Furthermore, RISC naturally addresses the challenge of collaborative critiquing through its intuitive approach to adapt items with similar components, efficiently leverage critiquing keyphrases for collaborative recommendation adjustments, and respond to users in real-time. Additionally, based on RISC, we designed a Weight Experience Replay method (WER) with a simplified loss function to reduce the computational overhead of replay. WER mitigates catastrophic forgetting by reusing a weighted average of historical samples, thereby deepening the model's image of users' stable preferences. In general, our contributions are presented as follows:

- We propose a generic RISC framework that indirectly estimates the relevance of the users to critiquing keyphrases through sampling important representative items, thereby empowering the majority of knowledge graph recommender models to refine recommendations efficiently.
- We present an experience replay method WER, which mitigates the forgetting of significant user preferences prior learned by the original model during the inference phase.
- Extensive experimentation was conducted on three benchmark datasets and three state-of-the-art knowledge graph recommendation algorithms. The results demonstrate the effectiveness and generalizability of RISC with WER.

2 PRELIMINARIES

2.1 Problem Formulation

In this section, we introduce the knowledge graph recommendation and formulate the critiquing task.

Knowledge Graph-based Recommendation. Let $\mathcal{U}$ be a set of users and $\mathcal{V}$ a set of items. Let $\mathcal{D}_{\mathcal{T}} = \{\langle u, v\rangle \,|u \in \mathcal{U}, v \in \mathcal{V}\}$ be a set of observed implicit feedback, where $\langle u, v\rangle$ pair indicates that user u has interacted with item i before. Let $\mathcal{G}$ be a knowledge graph containing a large number of semantic keyphrases, where keyphrases are entities in the KG. The objective of the knowledge graph recommendation system is, given $\mathcal{D}_{\mathcal{T}}$ and $\mathcal{G}$, to predict the probability of users interacting with items based on $p_t\left(\mathcal{D}_{\mathcal{T}}, \mathcal{G}|U, V, K\right)$, where U, V, and K represent the user embedding, item embedding, and keyphrase embedding, respectively. Additionally, it can provide recommendation explanations $\mathcal{K}_e$ for users from KG. For convenience, we use the function $\hat{y}\left(\langle u, v\rangle, U, V, K\right)$ to denote the assignment of predicting the relevance score of users to each item.

Critiquing Task Formulation. The critique-able recommender system offers users the opportunity to express their recognition of recommendations, including the option to critique keyphrases $k \in \mathcal{K}_e$. Specifically, based on the U, V, and K generated by the original KG recommender model, we aim to achieve an iterative critiquing mode for general KG recommender systems. It allows the user to interact with the system repeatedly and generates an updated user embedding U^* with the user critiquing data $\mathcal{D}_C^{+/-} = \{\langle u, k\rangle \,|u \in \mathcal{U}, k \in \mathcal{K}\}$ to renew scores for the candidate items and re-recommend top-k items, where the $\langle u, k\rangle$ pair indicates the user u has submitted keyphrase k and $+/-$ represents positive or negative user critiquing.

2.2 Comparisons to Previous Work

The research on critiquing is focused on conversational recommender systems [12, 16, 17, 44], where the iterative communication between the user and the system gradually refines the recommendations to meet the genuine needs of the user. However, the majority of these studies are conducted with a bespoke Variational Autoencoder (VAE) [1–3, 19, 22, 47]. They establish correlations between users and keyphrases in the implicit space by redesigning the encoder of the VAE in the training phase. Subsequently, the keyphrase embedding is updated through direct manipulation of the implicit features of keyphrases [17, 19] or the application of a

Bayesian approach [22, 47]. These models are all capable of embracing critique with keyphrases and decoding to generate adjusted recommendations. Recently, BCIE [28] attempted to combine KG to realize critiquing in CRS by employing a belief propagation approach to convey critiquing messages. This endeavor substantiates the assertion that KGs can prove instrumental in critiquing operations. While these methods demonstrate the capacity to adapt recommendations, they tend to exhibit suboptimal initial performance and limited utilization of the critiquing information. Moreover, these models are designed specifically for the critiquing task and are incapable of applying to traditional list-based recommendation scenarios as well as state-of-the-art methods.

The critiquing task bears resemblance to incremental learning in its formal description, but in practice they are quite distinct [20, 35]. Although the critiquing data streams are provided progressively one after another and suffer from the catastrophic forgetting problem, most of the incremental learning methods are unable to recognize user critiquing straightforwardly. They require coupling with a critiquing framework and proper adaptation in order to yield meaningful results. On the one hand, the optimization objective in the inference phase is not aligned with the training phase, unlike the current mainstream setup for incremental learning. This also signifies that most incremental learning methods, especially recommender incremental learning models, cannot handle the task of critiquing in recommendation [7, 41, 46]. On the other hand, while some recent work on recommender incremental learning has achieved notable success by contrasting learning or experience replay methods [40, 54], the majority of these methods introduce additional terms to the original loss function. However, given the necessity of responding to user critiques in a timely manner and the detrimental impact of the original loss function on the response time of the critiquing model, it is imperative to optimize the loss function for incremental learning to ensure a balance between efficacy and response time.

In conclusion, KG recommender algorithms have become one of the mainstream methods due to their superior performance. Furthermore, the utilization of KG enables better prehension of critiquing keyphrases. Accordingly, our objective is to develop a generic critique-able recommender paradigm based on KG recommender models that are adaptable to a broader range of recommendation scenarios, patching the weaknesses of previous approaches and enabling rapid and efficient adaptation to user feedback. Not only does it overcome many of the shortcomings of previous approaches, but it also integrates seamlessly with state-of-the-art methodologies.

3 METHODOLOGY

3.1 Knowledge graph-based Critiquing Framework

The overall objective of knowledge graph-based critiquing is to update recommendations based on the U, V, and K generated by the original KG recommender model and the user feedback. Referring to previous work and extended to KG recommendation, the model is required to update the user embedding according to any critiquing without disturbing the results of other users. Thus, the V and K of the original model need to be invariant. The updated user embedding U^* is obtained depending on $p\left(U^*|\mathcal{D}_C^{+/-}, V, K\right)$, according to the Bayesian formula:

$$\log p_c\left(U^*|\mathcal{D}_C^{+/-}, V, K\right) \propto \underbrace{p_c\left(\mathcal{D}_C^{+/-}|U^*, V, K\right)}_{\textit{critiquing likelihood}} \times \underbrace{p\left(U^*\right)}_{\textit{user prior}} \quad (1)$$

The first term in Eq. 1 is the critiquing likelihood, which describes the model updating its parameters after observing user feedback keyphrases. The second term is the user prior, which represents the empirical knowledge of the model regarding the user's preferences. Given the Maximum A Posteriori Estimation (MAP), the optimization objective is to maximize the user embedding posterior estimate:

$$U^* = \arg\max_{U^*}\left(\log p_c\left(\mathcal{D}_C^{+/-}|U^*, V, K\right) + \log p\left(U^*\right)\right) \quad (2)$$

Specifically, for critiquing likelihood, we expect to reasonably calculate the correlation of users for critiquing keyphrases to efficiently optimize the user embedding posterior. In the critiquing scenario, the keyphrase provides substantial indirect evidence of the user's preference. Considering the challenges of timeliness and effectiveness, we have developed a straightforward parameter update method to facilitate timely feedback. Furthermore, the extensive knowledge contained in the knowledge graph is employed to mine critiquing collaborative information, thereby enhancing the refinement of recommendations.

The user prior represents the user preferences that have been learned by the original model and we need to minimize the forgotten during the recommendation adjustment process. In essence, the prior can be described as the discrepancy between the prediction outcome of the recommender model on historical training data and the fundamental facts in the inference phase: $\log p\left(U^*\right) = \mathcal{L}\left(\mathcal{D}_\mathcal{T}, \Theta\right)$, where Θ denoting the parameters of the current model. Notice that the form of user prior is consistent with the recommended incremental learning. However, in critiquing, we expect to minimize the computational expense associated with preserving the user prior. Therefore, we simplify the general incremental learning method and use the strategy of experience replay to reserve the prior knowledge instead of maintaining the original loss function of continuous training. The subsequent section will analyze how to calculate the critiquing likelihood in a reasonable manner and preserve the user prior.

3.2 Sampling Representative Items for Critiquing

We denote the critiquing likelihood as the user's preference for keyphrase in $\mathcal{D}_C^{+/-}$:

$$\log p\left(\mathcal{D}_C^{+/-}|U^*, V, K\right) = \sum_{\langle u_i, k_j\rangle \in \mathcal{D}_C} \log p_c^{+/-}\left(\langle u_i, k_j\rangle\right) \quad (3)$$

where $p_c^{+/-}\left(\langle u_i, k_j\rangle\right)$ denotes the estimate of the relevance of user u_i to keyphrase k_j. In previous studies, researchers have developed specific modules during the training phase to intuitively portray user preferences for keyphrases. However, determination of $p_c^{+/-}\left(\langle u_i, k_j\rangle\right)$ is challenging in the most KG recommender methods since they do not establish a verifiable connection between

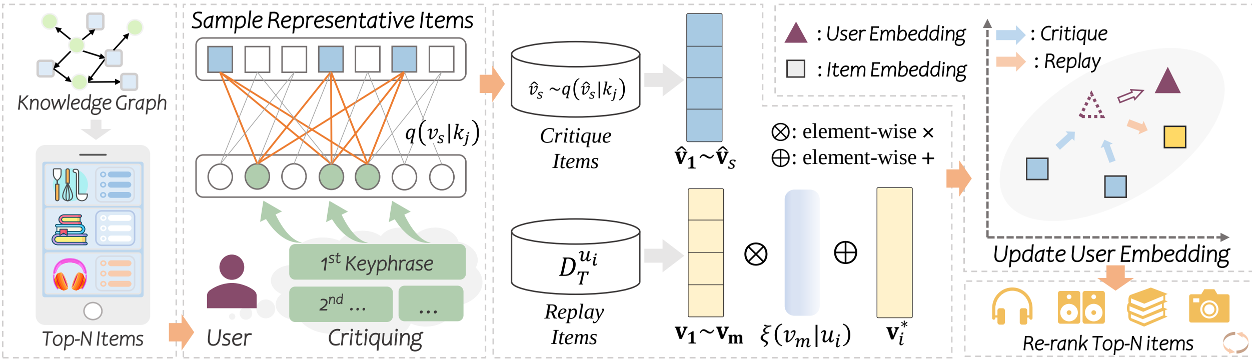

Figure 2: The overall framework of our proposed RISC and WER, which leverages knowledge graph sampling the importance representatives and implement weight experience replay refine recommendations according to user critiquing.

the users and the keyphrases. Instead, the model is optimized by aggregating items' neighbor nodes using the KG topological information during the training stage. Inspired by the aforementioned aggregation, we put forth a generic Sampling Representative Items method for Critiquing on KG recommender methods (RISC), which conversely treats the items node in the KG as a hinge to propagate information from critiquing keyphrases to users, thereby indirectly estimating the relevance of the users towards the keyphrases. Given that each node in the knowledge graph has a large number of neighbors, it is impractical to include all these neighbors in the inference procedure. Consequently, we determine in sampling representative items as the proxy for critiquing keyphrases. Formally, $p_c^{+/-}\left(\langle u_i, k_j\rangle\right)$ is transformed into the product of the score between user and proxy $\bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right)$ and the representative coefficient between v_s and k_j, $p_r\left(v_s|k_j\right)$. To facilitate the calculation, the Jensen inequality is employed to convert the optimization objective into maximizing the lower bound of the critiquing likelihood:

$$\begin{aligned}
&\log p\left(\mathcal{D}_C^{+/-}|U^*, V, K\right) \\
&\quad= \sum_{\langle u_i,k_j\rangle\in\mathcal{D}_C^{+/-}} \log \sum_{v_s\in V} \bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right)\cdot p_r\left(v_s|k_j\right) \\
&\quad= \sum_{\langle u_i,k_j\rangle\in\mathcal{D}_C^{+/-}} \log\mathbb{E}_{v_s\sim p_r\left(v_s|k_j\right)}\bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right) \\
&\quad\geqslant \sum_{\langle u_i,k_j\rangle\in\mathcal{D}_C^{+/-}} \mathbb{E}_{v_s\sim p_r\left(v_s|k_j\right)}\log \bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right)
\end{aligned} \tag{4}$$

Since V and K remains invariant, for simplicity we concisely denote the $\hat{y}\left(\langle u_i, v_s\rangle, U^*, V, K\right)$ as $\hat{y}\left(\langle u_i, v_s\rangle, U^*\right)$ below. And given $\bar{p}_c^{+}\left(\langle u_i, v_s\rangle\right) = \hat{y}\left(\langle u_i, v_s\rangle, U^*\right)$ denotes the user positive critiquing, and $\bar{p}_c^{-}\left(\langle u_i, v_s\rangle\right) = 1-\hat{y}\left(\langle u_i, v_s\rangle, U^*\right)$ indicates the negative. Meanwhile, sampling important representative items also intuitively satisfies our goal of mining collaboratively critiquing items with content similar to keyphrases using knowledge graph, sufficiently exploiting the user feedback. Nevertheless, there are variety of ways to learn KG structural information in different original models, which makes it difficult to compute the coefficient between representative items and critiquing keyphrases straight away. As a result, we apply importance sampling to the expectation $\mu_p = \mathbb{E}_{v_s\sim p_r\left(v_s|k_j\right)}\log \bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right)$, which gives:

$$\mu_q = \mathbb{E}_{v_s\sim q\left(v_s|k_j\right)} \frac{p_r\left(v_s|k_j\right)}{q\left(v_s|k_j\right)} \log \bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right) \tag{5}$$

where $q\left(v_s|k_j\right)$ is defined as the auxiliary probability of sampling according to keyphrase k_j. To reduce the computational complexity of the inference phrase and accelerate the calculation, we approximate the expectation μ_q using the Monte Carlo method:

$$\hat{\mu}_q = \frac{1}{N}\sum_{s=1}^{N} \frac{p_r\left(\hat{v}_s|k_j\right)}{q\left(\hat{v}_s|k_j\right)} \log \bar{p}_c^{+/-}\left(\langle u_i, \hat{v}_s\rangle\right), \hat{v}_s \sim q\left(\hat{v}_s|k_j\right) \tag{6}$$

where $\hat{\mu}_q$ is an unbiased estimate of μ_q. The remaining question is how we define the supplementary sampler $q\left(v_s|k_j\right)$, which should reduce the variance induced by importance sampling. Excessive variance can prevent effective training. From the derivation of the importance sample in [6], we can give the variance:

$$\begin{aligned}
&\mathrm{Var}\left(\hat{\mu}_q\right) \\
&= \frac{1}{n}\left(\mathbb{E}_{v_s\sim q\left(v_s|k_j\right)}\left(\left(\frac{\log \bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right)\cdot p_r\left(v_s|k_j\right)}{q\left(v_s|k_j\right)}\right)^2\right) - \mu_p^2\right)
\end{aligned} \tag{7}$$

Therefore, the optimal sampler that minimizes the variance $\mathrm{Var}\left(\hat{\mu}_q\right)$ in Eq.7 is given by:

$$q^* = \frac{p_r\left(v_s|k_j\right)\cdot\left|\log \bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right)\right|}{\sum_{s=1}^{N} p_r\left(v_s|k_j\right)\cdot\left|\log \bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right)\right|} \tag{8}$$

Unfortunately, in RISC, the probability of representing the relationship between items and keyphrases cannot be measured uniformly, making it difficult to calculate an optimal sampler. Considering that these methods learn more about the topological structure of the knowledge graph, we replace $p_r\left(v_s|k_j\right)$ by applying a function similar to the attention mechanism:

$$\begin{aligned}
q^* &= \frac{\alpha\left(v_s, k_j\right)\cdot\left|\log \bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right)\right|}{\sum_{s=1}^{N} \alpha\left(v_s, k_j\right)\cdot\left|\log \bar{p}_c^{+/-}\left(\langle u_i, v_s\rangle\right)\right|} \\
\alpha\left(v_s, k_j\right) &= \frac{\exp\left(\sigma\left(v_s\cdot k_j\right)\right)}{\sum_{v_{s'}\in H\left(k_j\right)}\exp\left(\sigma\left(v_{s'}\cdot k_j\right)\right)}
\end{aligned} \tag{9}$$

The $H(k_j)$ represents the first-order items neighbors set of k_j, $\alpha(v_s, k_j)$ denotes the representative coefficient between k_j and v_s, and σ is the sigmoid function. While we could not claim that $\alpha(v_s, k_j)$ properly fits arbitrary recommendation methods, importance sampling can effectively reduce the sample size required and the variance of the estimation using the Monte Carlo method.

3.3 Experience Reply in Multi-step Critiquing

During the inference phase, the user-system interaction undergoes a constantly evolving process. In section 3.2, we discussed the methodology for refining recommendations based on critiquing. However, when faced with a continuous stream of user feedback, recommendation performance may collapse after multiple critiquing. The reason is that during the continuous fine-tuning of the user embedding, important features learned by the original model may be forgotten, resulting in catastrophic forgetting. Consequently, Eq. 2 attempts to deepen the knowledge learned from the original model in inference phrase. Following the experience replay strategy, we can explicitly maintain a reservoir to buffer a small subset of historical data, which serves to alleviate the tendency to forget previously learned knowledge.

Based on RISC, the critiquing target $\langle u, k\rangle$ pair changes from to $\langle u, v\rangle$ pair, which is consistent with the training stage. Using the loss function of the original model as $\mathcal{L}_{prior}$ is the most direct way. However, this will inevitably result in an increased cost and a notable impact on the response speed because of the complicated design of original model. We want a simple and efficacious way to reproduce historical data that remembers old user perferences. Given that the loss function of the original model is frequently constituted of multiple components, we have selected the recommender loss from these methods as the replay loss:

$$\log p\left(U^*\right) = \sum_{(u_i,v_j)\in\mathcal{D}^*_{\mathcal{T}}} \sum_{\left(u_i,v'_j\right)\notin\mathcal{D}^*_{\mathcal{T}}} \log\left(\hat{y}\left(\left\langle u_i, v_j\right\rangle, U^*\right) - \hat{y}\left(\left\langle u_i, v'_j\right\rangle, U^*\right)\right) \tag{10}$$

where $\mathcal{D}^*_{\mathcal{T}} \subseteq \mathcal{D}_{\mathcal{T}}$ denotes the reservoir. The most influential data in the buffer determines replay quality and model performance. In incremental learning, previous research has demonstrated that employing the nearest-mean-of-exemplars classification rule is not only computationally straightforward but also effective in avoiding catastrophic forgetting [20]. In light of the necessity to hold the items embedding invariant with the function outlined in Eq. 10, we posit that the mean-of-exemplars itself is an optimal replay object within critiquing scenario, as it relies on the storage of a single item for each user in the reservoir. Specifically, the utilization of the mean-of-exemplars replay can be abstracted as follows:

$$U^* = \arg\min_{U^*} \sum_{(u_i,v_j)\in\mathcal{D}^{u_i}_{\mathcal{T}}} \left(\hat{y}\left(\langle u_i, \bar{v}\rangle, U^*\right) - \hat{y}\left(\langle u_i, v_j\rangle, U\right)\right) \tag{11}$$

, where $\bar{v}_i = \frac{1}{|\mathcal{D}^{u_i}_{\mathcal{T}}|} \sum_{(u_i,v_j)\in\mathcal{D}^{u_i}_{\mathcal{T}}} v_j$. While mean-of-exemplars considers the entirety of the training set, it fails to account for the disparate influence of individual items. More powerful items can reinforce the model's memory of stable user preferences than the less frequently considered, e.g., long-tail items. Hence, we present weight experience replay (WER), a method that aims to amplify the influence of these items and assign weights to different items to generate new replay examples:

$$v^*_i = \sum_{(u_i,v_j)\in\mathcal{D}^{u_i}_{\mathcal{T}}} \xi\left(v_j|u_i\right)\cdot v_j \tag{12}$$

Since the original model incorporates the aggregation of side information from KG, we consider using KG to calculate the weights of different items would be beneficial. To elaborate, the subgraph comprising the items that the user has interacted with in $\mathcal{D}_{\mathcal{T}}$ and their neighbors can be regarded as a representation of the user, which is referred to as the user subgraph. We assume that the weight of each item on the user can be determined by the similarity between the item neighbor subgraph and the user subgraph. We use Jaccard coefficient to calculate the weight: $\xi\left(v_j|u_i\right) = \frac{|A(u_i)\cap S(v_j)|}{|A(u_i)\cup S(v_j)|}$, $A(u_i) = S(v_1)\cup\cdots\cup S(v_m)$, $v_m \in \mathcal{D}^{u_i}_{\mathcal{T}}$, where $S(\cdot)$ refers to the set of neighbors of a node. Ultimately, the replay loss is given by:

$$\log p\left(U^*\right) = \sum_{(u_i,v'_j)\notin\mathcal{D}_{\mathcal{T}}} \log\left(\hat{y}\left(\left\langle u_i, v^*_i\right\rangle, U^*\right) - \hat{y}\left(\left\langle u_i, v'_j\right\rangle, U^*\right)\right) \tag{13}$$

Update embedding Finally, the maximum posteriori probability of the user embedding can be transformed to minimize the loss function. Making $\hat{y}(\langle u, v\rangle, U^*) = \sigma(U^*[u]^\top \cdot V[v])$, and for brevity, we simply denote the $U^*[u]$ as $\overset{*}{\mathbf{u}}$ and the $V[v]$ as $\mathbf{v}$. For the proposed RISC and WER, we have the following loss function:

$$\begin{aligned}
&\min \mathcal{L}_{RISC+WER} \\
&= -\log\left(p_c\left(\mathcal{D}^{+/-}_C|U^*, V, K\right)\cdot p\left(U^*\right)\right) \\
&= -\eta\cdot\frac{1}{N}\sum_{\langle u_i,k_j\rangle\in\mathcal{D}^+_C}\sum_{s=1}^{N}\left(\log\sigma\left(\overset{*}{\mathbf{u}}{}_i^\top\cdot\hat{\mathbf{v}}_s\right) + \log\sigma\left(\overset{*}{\mathbf{u}}{}_i^\top\cdot\mathbf{v}^*_i\right)\right), or \\
&= -\eta\cdot\frac{1}{N}\sum_{\langle u_i,k_j\rangle\in\mathcal{D}^-_C}\sum_{s=1}^{N}\left(\log\left(1-\sigma\left(\overset{*}{\mathbf{u}}{}_i^\top\cdot\hat{\mathbf{v}}_s\right)\right) + \log\sigma\left(\overset{*}{\mathbf{u}}{}_i^\top\cdot\mathbf{v}^*_i\right)\right)
\end{aligned} \tag{14}$$

where η is the learning rate. This formula excludes the random negative sampling term in the user prior and substitutes it with 0. That's not affect the training and reduces the computational cost. Experiments have also validated this statement.

4 EXPERIMENTS

We present empirical results to substantiate the efficacy of the proposed RISC with WER. The experiment is intended to answer the following research questions.

RQ1: How does the performance of RISC with WER in refining recommendations according to user critiquing compare to a diverse range of baselines? Does it generalize to the knowledge graph recommender models? Does WER have advantages over other incremental learning methods in the RISC framework?

RQ2: Can sampling important representatives extract critiquing information more productively? Does the WER can genuinely mitigate the issue of catastrophic forgetting?

RQ3: What is the effect of varying number of samples on RISC? Can the experiment give an intuitive visualization of the critiquing task?

4.1 Experiment Settings

We validate all methods on three real-world datasets: MovieLens-1M, Last-FM, and Alibaba-iFashion. In the inference phase, we simulate a total of 10 rounds of critiquing tasks, adopting some regular settings from previous research, and each round represents a new critiquing from users [22, 47]. In order to simulate authentic 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 RISC, we calculated the differences between the average frequency of keyphrases occurrences corresponding to the Top-10 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.

Two metrics, NDCG@5 and HR@5, were employed in a multi-step critiquing to determine the performance of the models in enhancing recommendations based on user feedback. It is important to note that meaningful comparisons of curve trends can only be conducted if the starting scores are approximately the same, as previously discussed in the literature. However, as illustrated in Table 1, the base recommender scores of the critiquing baselines are markedly inferior to those KG approaches. They are therefore unsuitable for use in general KG recommender models. To ensure fair comparisons, we have re-implemented their methodology in the KG recommender system, although this may cause potential issues. Besides, a more detailed description of the datasets, evaluation criteria, and experimental settings can be found in the appendix.

4.2 Overall Performance Comparison (RQ1)

For a comprehensive evaluation, we will first introduce three state-of-the-art KG recommender models KGAT [37], KGIN [38], and DiffKG [10]. We then use the NDCG@5 (HR@5 is reported in the Appendix) trend over multiple rounds of critiquing to compare RISC+WER with two different types of state-of-the-art baselines: the critiquing methods CE-VAE [17], BK-VAE [47], DCE-VAE [22], and BCIE [28]; and the recommender incremental learning methods iCaRL [20], lwcKD [40], and INFER [54], respectively. We describe these methods and how we applied them in KG critiquing scenarios in more detail in the Appendix.

- The base recommendation rating of diverse models is illustrated in Table 1, which demonstrates that all three KG recommendation methods are more competitive than the specially designed critiquing methods. A superior original model provides users with an optimal experience while improving the ability of the more efficacious models is evidently more challenging.
- Evaluations of the performance of all methods on three datasets consistently indicate that RISC+WER outperforms all baseline methods. Although the experiments simulate an ideal condition, the superiority of RISC+WER in enhancing recommendations and preserving previously learned knowledge is evident from the results of the continuous 10-step critiquing task. Specifically, the meticulously crafted important sampling framework capitalizes on user critiquing and expands keyphrase collaborative information to refine recommendations through KG rationally. Concurrently, the WER significantly strengthens the memory of the user stable perferences, thus producing the phenomenon of a continuously ascending curve in the experiments.
- The CE, BK, DCE, and BCIE all demonstrate suboptimal results. This is due to their direct optimization of the correlation of users to keyphrases, which is beneficial in specially designed critiquing models but has minimal impact on the mostly KG recommender models. Some methods have been found to have a negative effect. Additionally, all of these methods have been observed to exhibit the typical characteristics of catastrophic forgetting and have been found to confirm the capability of RISC+WER.
- The line trend of iCaRL, lwcKD, and INFER unambiguously reveals distinctions between the critiquing and the incremental learning task. While these incremental learning methods show a slight tendency to refine recommendation capability by sustaining prior knowledge in some cases, the obvious discrepancies with the RISC+WER illustrate the difficulty of using incremental learning methods in isolation to deal with the form of critiquing data streams, which highlights the advantage of the RISC in exploring user feedback keyphrases through KG.

Table 1: Performance of different original methods. The metrics are NDCG@5 and HR@5.

	MovieLens		Last-FM		Alibaba	
	ndcg	hr	ndcg	hr	ndcg	hr
CE-VAE	0.0680	0.1701	0.0191	0.0379	0.0103	0.0387
BK-VAE	0.0793	0.2037	0.0232	0.0462	0.0134	0.0497
DCE-VAE	0.0842	0.2171	0.0231	0.0463	0.0138	0.0515
BCIE	0.0690	0.1559	0.0397	0.0745	0.0196	0.0722
KGAT	0.1967	0.5172	0.0612	0.1789	0.0356	0.1174
KGIN	0.2687	0.6368	0.0743	0.1960	0.0460	0.1486
DiffKG	0.2589	0.6322	0.0843	0.2153	0.0467	0.1490

4.3 Performance Comparison of Incremental Learning on RISC (RQ1)

The experience reply is essentially decoupled from the process of updating user embedding posteriors. To assess the proposed WER in capturing old user preferences, we investigate the impact of different incremental learning methods within the RISC framework. Table 2 depicts the final performance of different incremental learning methods on the Last-FM dataset following ten rounds of critiquing. Table 3 illustrates the time required to complete one round of training for each method. The results are averaged 5 repeated experiments, where the best performance is denoted in bold.

Firstly, we can observe that all the continuous learning methods in the RISC framework demonstrate an excellent ability to adjust recommendations based on user critiquing, which proves our previous point. And INFER, which finds the most influential examples, shows better results than the nearest-mean method in most

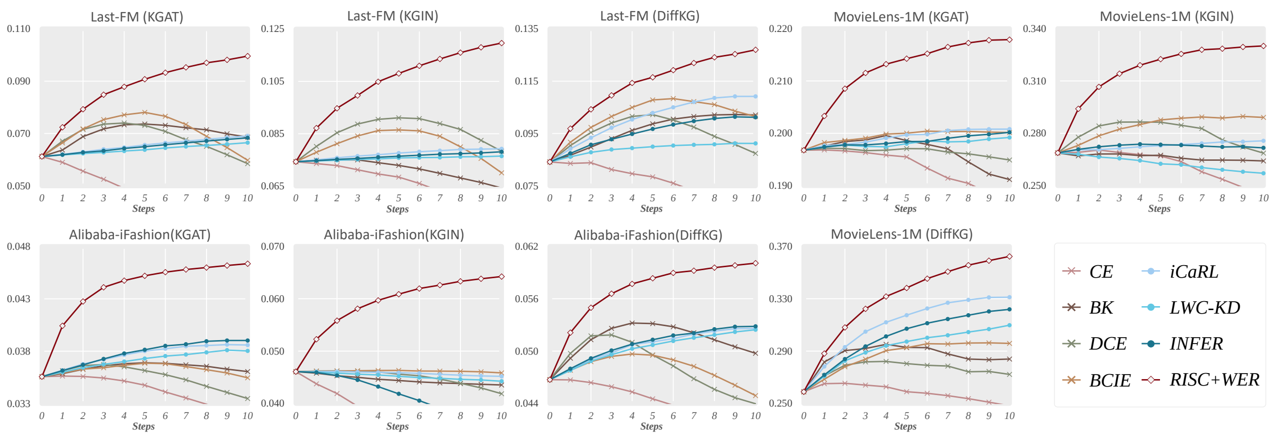

Figure 3: Comparison of RISC+WER and baseline methods for *NDCG@5* on Last-FM, Alibaba-iFashion and MovieLens, where the original models are KGAT ,KGIN and DiffKG.

Table 2: Performance Comparison of Incremental Learning Baselines within RISC. The metrics are NDCG@5 and HR@5.

RISC+/	KGAT NDCG	KGAT HR	KGIN NDCG	KGIN HR	DiffKG NDCG	DiffKG HR
Last-FM						
iCaRL	0.0938	0.2530	0.1160	0.2892	0.1268	0.3088
lwcKD	0.0885	0.2377	0.1047	0.2634	0.1083	0.2723
INFER	0.0995	0.2612	0.1162	0.2870	0.1250	0.3056
WER	**0.0996**	**0.2625**	**0.1196**	**0.2957**	**0.1270**	**0.3114**
Alibaba-iFashion						
iCaRL	0.0457	0.1439	0.0622	0.1838	0.0589	0.1779
lwcKD	0.0461	0.1449	0.0626	0.1834	0.0584	0.1768
INFER	0.0460	0.1452	0.0629	0.1844	0.0593	0.1791
WER	**0.0464**	**0.1452**	**0.0642**	**0.1872**	**0.0601**	**0.1804**
MovieLens-1M						
iCaRL	0.2159	0.5233	0.3216	0.6725	0.3407	0.7238
lwcKD	0.2152	0.5276	0.3229	0.6814	0.3429	0.7250
INFER	0.2168	0.5279	0.3226	0.6741	0.3535	0.7265
WER	**0.2179**	**0.5316**	**0.3301**	**0.6817**	**0.3623**	**0.7380**

situations. Although the difference in performance between these methods is insignificant, WER performs better than other incremental learning methods.That's because WER considers all user data for replay and leverages a knowledge graph to maintain stable user preferences. Additionally, its smaller pool of replay allows for faster recommendation adjustments.

Table 3: Comparison of time overhead and reservoir size for different incremental learning baselines.

RISC+/	reservoir	MovieLens	Last-FM	Alibaba
iCaRL	10	21s	51s	1.1s
lwcKD	-	67s	86s	4s
INFER	10	22s	59s	1.2s
WER	1	10s	33s	0.6s

4.4 Effect of Important Sampling (RQ2)

This study aims to ascertain the impact of importance sampling and the role of knowledge graph in selecting important representatives. For intuitive illustration, we elected to compare the first-round critiquing ratings of importance sampling, random sampling, and not utilizing KG sampling. The number of representatives sampled for each keyphrase is fixed at 10. The results of the experiment on the Last-FM are shown in Table 4. Results indicate that the importance sampling method is indeed superior to the random sampling method, which shows that sampling important representatives can more fully exploit collaborative critique information to tune recommendations efficiently. In contrast, the observed increase in the experiment without using KG is minimal, which highlights the importance of KG in selecting representative items.

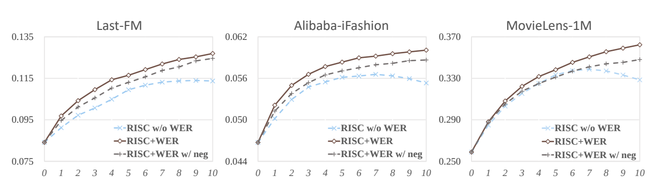

Figure 4: Illustration of the experiment results of exploring the effect of WER on three datasets, where the original model is DiffKG and the metric is NDCG@5.

Table 4: Comparison of the performance on Last-FM of different sampling methods after the first step critiquing. The metrics are NDCG@5 and HR@5.

Last-FM (1st step critiquing)						
	KGAT		KGIN		DiffKG	
	NDCG	HR	NDCG	HR	NDCG	HR
Intial rating	0.0612	0.1789	0.0743	0.1960	0.0843	0.2153
Sample w/o KG	0.0614	0.1792	0.0742	0.1957	0.0842	0.2153
Random sample	0.0692	0.1986	0.0824	0.2141	0.0918	0.2360
RISC	0.0724	0.2072	0.0872	0.2271	0.0969	0.2473

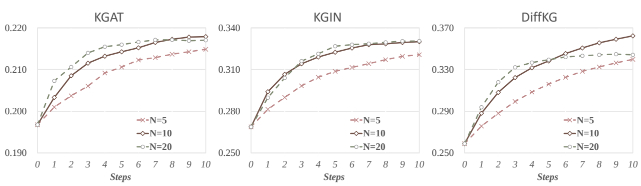

Figure 5: Illustration of the investigation of the number of importance sampling on MovieLens-1M, where the metric is NDCG@5.

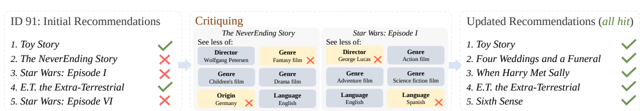

Figure 6: An example of the whole process of user Id: 91 critiquing.

4.5 Effect of Experience Replay (RQ2)

As observed in Section 4.2, the critiquing methods obviously suffered catastrophic forgetting, while the incremental learning methods appeared to resist forgetting somewhat, which suggests that incremental learning methods may offer a valuable contribution to multi-step critiquing tasks. The combination of RISC and WER demonstrated the power of steady adaptation recommendations, although it was unclear whether this was due to the RISC itself or the WER method. A further investigation was therefore conducted to ascertain the specific effects of the experience replay method. As shown in Figure 4, the absence of WER is evident in the observation that, despite the RISC's notable tuning capability, the recommended performance declines after multiple iterations of critiquing. It should be noted that WER does serve to preserve prior knowledge of user preferences and provides powerful assistance in preventing catastrophic forgetting. In addition, we have also experimentally verified the effect of adding randomly sampled negative examples to the WER for training, as represented by the grey curve in Figure 4. Seeing the negative items actually interferes with the model's ability to resist forgetting. This might be because these negative examples erroneously prompt the model to capture the user's old preferences, reducing the model's performance.

4.6 Impact of Sampling Number (RQ3)

The quality of the important sampling representatives for critiquing keyphrases determines the effectiveness of refined recommendations. According to the Monte Carlo method, a greater quantity of samples will produce superior results. However, an excessive number of samples would result in unnecessary expenditure of time and space. Identifying the optimal number of samples would be beneficial to achieve an effective balance between performance and overhead. We exposed the number of samples in the Movielens-1M dataset in N = {5, 10, 20}. As shown in Figure 5, a small sample size of N=5 indicates that the representative items cannot fully leverage the critiquing collaborative information. While the sample size of N=20 may appear more impressive, it does not match the significant performance improvement and shows a decline in model performance towards the end of the multi-step critiquing process. The impact of varying the number of samples on the model is negligible. Consequently, N=10 represents the optimal choice.

4.7 Case Study (RQ3)

We illustrate the critiquing process using a randomly selected user with ID 91 from the Movielens dataset. As shown in Figure 6, the initial recommendation includes only two movies he is interested in, "Toy Story" and "E.T. the Extra-Terrestrial". Subsequently, he selected four items based on the system's critiquing interface, namely preferences, namely "Fantasy," "Germany," "George Lucas," and "Spanish". The recommendation system then updated his recommendations based on these keyphrases, successfully aligning them with the user's interests. This example demonstrates that RISC+WER is effective in refining recommendations based on initial recommendations that already have a high degree of accuracy.

5 CONCLUSION

This research proposes RISC and WER, a general plugin acting on most KG recommendation models, to provide new options for refining recommendations through critiques in mainstream recommendation scenarios. The framework employs a knowledge graph to suggest a method for sampling important representative items as a hinge for transferring information and mining critiquing collaborative information. It also designs the KG-based experience replay method WER to strengthen stable user preference. Extensive evaluations on benchmark datasets and KG recommendation models demonstrate that our proposed RISC+WER framework can operate with different KG model architectures and mitigate catastrophic forgetting. Furthermore, it demonstrates a notable improvement in recommendation refinement compared to various baselines.

Furthermore, our work is limited as it only concentrates on KG recommender models. In the future, we aim to broaden the scope of critiquing and the development of more efficient methods for linking users with relevant keyphrases and identifying items that users are uninterested in.

REFERENCES

[1] Diego Antognini and Boi Faltings. 2021. Fast multi-step critiquing for VAE-based recommender systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 209–219.

[2] Diego Antognini and Boi Faltings. 2022. Positive and Negative Critiquing for VAE-based Recommenders. *arXiv preprint arXiv:2204.02162* (2022).

[3] Diego Antognini, Claudiu Musat, and Boi Faltings. 2020. Interacting with explanations through critiquing. *arXiv preprint arXiv:2005.11067* (2020).

[4] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. 2020. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems* 33 (2020), 15920–15930.

[5] Luke Friedman, Sameer Ahuja, David Allen, Terry Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, et al. 2023. Leveraging Large Language Models in Conversational Recommender Systems. *arXiv preprint arXiv:2305.07961* (2023).

[6] John Hammersley. 2013. *Monte carlo methods*. Springer Science & Business Media.

[7] Bowei He, Xu He, Yingxue Zhang, Ruiming Tang, and Chen Ma. 2023. Dynamically expandable graph convolution for streaming recommendation. In *Proceedings of the ACM Web Conference 2023*. 1457–1467.

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

[9] Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2023. Recommender ai agent: Integrating large language models for interactive recommendations. *arXiv preprint arXiv:2308.16505* (2023).

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

[11] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[12] Hanze Li, Scott Sanner, Kai Luo, and Ga Wu. 2020. A ranking optimization approach to latent linear critiquing for conversational recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 13–22.

[13] Qingfeng Li, Huifang Ma, Ruoyi Zhang, Wangyu Jin, and Zhixin Li. 2023. Dual-view co-contrastive learning for multi-behavior recommendation. *Applied Intelligence* (2023), 1–18.

[14] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 305–314.

[15] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems* 30 (2017).

[16] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. 2020. Latent linear critiquing for conversational recommender systems. In *Proceedings of The Web Conference 2020*. 2535–2541.

[17] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep critiquing for VAE-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 1269–1278.

[18] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *The world wide web conference*. 1210–1221.

[19] Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. 2021. Disentangling preference representations for recommendation critiquing with ß-vae. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1356–1365.

[20] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2001–2010.

[21] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in neural information processing systems* 32 (2019).

[22] Tianshu Shen, Zheda Mai, Ga Wu, and Scott Sanner. 2022. Distributional Contrastive Embedding for Clarification-based Conversational Critiquing. In *Proceedings of the ACM Web Conference 2022*. 2422–2432.

[23] Xiaoxuan Shen, Baolin Yi, Hai Liu, Wei Zhang, Zhaoli Zhang, Sannyuya Liu, and Naixue Xiong. 2019. Deep variational matrix factorization with knowledge embedding for recommendation system. *IEEE Transactions on Knowledge and Data Engineering* 33, 5 (2019), 1906–1918.

[24] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. 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*. 528–536.

[25] Yubo Shu, Hansu Gu, Peng Zhang, Haonan Zhang, Tun Lu, Dongsheng Li, and Ning Gu. 2023. RAH! RecSys-Assistant-Human: A Human-Central Recommendation Framework with Large Language Models. *arXiv preprint arXiv:2308.09904* (2023).

[26] Rui Sun, Xuezhi Cao, Yan Zhao, Junchen Wan, Kun Zhou, Fuzheng Zhang, Zhongyuan Wang, and Kai Zheng. 2020. Multi-modal knowledge graphs for recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1405–1414.

[27] Zhenwei Tang, Griffin Floto, Armin Toroghi, Shichao Pei, Xiangliang Zhang, and Scott Sanner. 2023. LogicRec: Recommendation with Users' Logical Requirements. *arXiv preprint arXiv:2304.11722* (2023).

[28] Armin Toroghi, Griffin Floto, Zhenwei Tang, and Scott Sanner. 2023. Bayesian Knowledge-driven Critiquing with Indirect Evidence. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1838–1842.

[29] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. 2018. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 592–600.

[30] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 417–426.

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

[32] 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 *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 968–977.

[33] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *The world wide web conference*. 2000–2010.

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

[35] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[36] Wenjie Wang, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2022. User-controllable recommendation against filter bubbles. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1251–1261.

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

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

[39] Xiting Wang, Kunpeng Liu, Dongjie Wang, Le Wu, Yanjie Fu, and Xing Xie. 2022. Multi-level recommendation reasoning over knowledge graphs with reinforcement learning. In *Proceedings of the ACM Web Conference 2022*. 2098–2108.

[40] Yuening Wang, Yingxue Zhang, and Mark Coates. 2021. Graph structure aware contrastive knowledge distillation for incremental learning in recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3518–3522.

[41] Yuening Wang, Yingxue Zhang, Antonios Valkanas, Ruiming Tang, Chen Ma, Jianye Hao, and Mark Coates. 2023. Structure aware incremental learning with personalized imitation weights for recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4711–4719.

[42] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. 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*. 219–228.

[43] Zhen Wang, Liu Liu, Yiqun Duan, Yajing Kong, and Dacheng Tao. 2022. Continual learning with lifelong vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 171–181.

[44] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep language-based critiquing for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 137–145.

[45] Lianghao Xia, Chao Huang, Jiao Shi, and Yong Xu. 2023. Graph-less collaborative filtering. In *Proceedings of the ACM Web Conference 2023*. 17–27.
[46] Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark Coates. 2020. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2861–2868.
[47] Hojin Yang, Tianshu Shen, and Scott Sanner. 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*. 2111–2115.
[48] 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*. 3046–3056.
[49] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1434–1443.
[50] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*. PMLR, 3987–3995.
[51] Mengyao Zhai, Lei Chen, Frederick Tung, Jiawei He, Megha Nawhal, and Greg Mori. 2019. Lifelong gan: Continual learning for conditional image generation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2759–2768.
[52] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.
[53] Huanyu Zhang, Xiaoxuan Shen, Baolin Yi, Wei Wang, and Yong Feng. 2023. KGAN: Knowledge grouping aggregation network for course recommendation in MOOCs. *Expert Systems with Applications* 211 (2023), 118344.
[54] Xinni Zhang, Yankai Chen, Chenhao Ma, Yixiang Fang, and Irwin King. 2024. Influential Exemplar Replay for Incremental Learning in Recommender Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9368–9376.
[55] Xiaoyu Zhang, Xin Xin, Dongdong Li, Wenxuan Liu, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. 2023. Variational Reasoning over Incomplete Knowledge Graphs for Conversational Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 231–239.
[56] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 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*. 1006–1014.
[57] 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 *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1358–1368.
[58] Ding Zou, Wei Wei, Ziyang Wang, Xian-Ling Mao, Feida Zhu, Rui Fang, and Dangyang Chen. 2022. Improving knowledge-aware recommendation with multi-level interactive contrastive learning. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 2817–2826.

A RELATED WORK

Knowledge Graph Recommendations have emerged as one of the most popular and well-performing models in the field of recommender systems, offering a powerful way to enhance recommendations by utilizing a large amount of knowledge in KG. Early approaches like CKE [52] and DKN [31] leveraged pre-training on KG to strengthen item embeddings for improved recommendations [29]. In recent years, end-to-end models based on graph convolution [34, 37, 38] or graph attention networks[30, 32, 42, 53] have proven more effective in mining information from KG, which are state-of-the-art solutions that exhibit strong competitiveness. In addition, a multimodal knowledge graph has been attempted to be applied [26], and there are also methods that try to reduce the noise from higher-order information [13]. Some methods [33] consider the coupling between the KG representation task and the recommendation task, incorporate collaborative information [42], 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 [48, 49, 57, 58] or used data augmentation methods [10] to optimize knowledge graphs, 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 a traditional recommendation system in that it requires the modeling of not only user preferences for items but also the correlation between users and keyphrases [1–3, 19, 22, 47]. 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 [5, 9, 55, 56]. However, most of these approaches are implemented within specially designed frameworks in order to model the association between users and keyphrases. Some VAE-based [14, 23, 24] methods encode the association between users and keyphrases into the latent space by redesigning the encoder during the training phase. This allows the latent state of keyphrases to be directly manipulated in the latent space during subsequent inference [17, 19]. Alternatively, the generated items and user embedding can be influenced by updating the posterior of keyphrase feature representation through Bayesian methods [22, 47]. These models are designed to accept feedback on keyphrases, allowing for the decoding of adjusted recommendation results. Moreover, BCIE [28] is, to our knowledge, the first method to attempt to incorporate KG into CRS to implement critiquing. It represents user preferences in the form of triples using knowledge representation methods and implements the transfer of critiquing information using belief propagation methods. It effectively utilizes the topological structure information of the KG to demonstrate that KG can play a role in the critiquing task. While these methods demonstrate the ability to adjust recommendations based on user feedback, they are designed specifically for the critiquing task, resulting in poor initial recommendation performance and low utilization of keyphrases. Nonetheless, they are incapable of being employed in traditional list-based recommendation scenarios or state-of-the-art methods.

Incremental learning provides intelligent models with the ability to continuously learn new knowledge [4, 20, 35]. The main challenge is that adapting to new data can lead to a decrease in the ability to capture the old distribution, i.e., catastrophic forgetting. The Incremental learning (IL) method attracted significant interest from researchers in the field of computer vision [43, 50]. Over time, it has been adopted in a growing number of sectors. According to mainstream research, IL methods can generally be divided into three categories. The first category is the regularization-based approach [11, 51], which is characterized by adding explicit regularization terms to balance the old and new tasks. The second category is the optimization-based approach [15], which generally explicitly designs and manipulates the optimization programs.

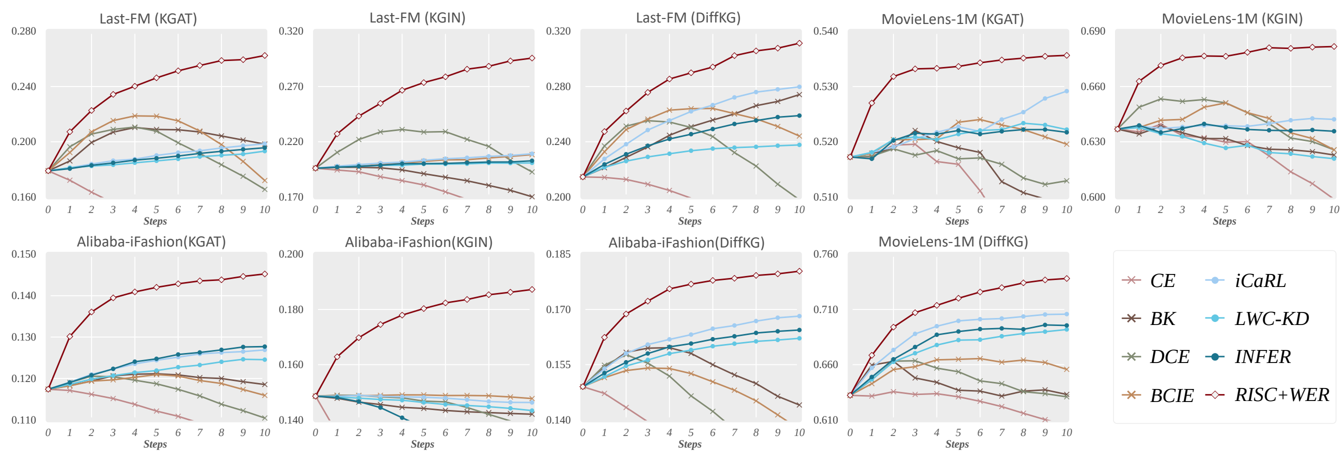

Figure 7: Comparison of RISC+WER and baseline methods for *HR@5* on Last-FM, Alibaba-iFashion and MovieLens, where the original models are KGAT ,KGIN and DiffKG.

The third category is the replay-based approach [21], which implements incremental learning by approximating and recovering old data distributions. In addition, most of the above methods are designed for image processing tasks. Recently, a few attempts have been made to apply incremental learning to recommender systems [7, 40, 41, 46, 54]. They achieved some results by comparing learning optimization programs with experience replay methods.

B EXPERIMENTS

B.1 Dataset Description

We validate IPGC on three real-world datasets for movie, music, and business in the experiments: (1) We use the Movielens-1M, a widely used movie benchmark dataset released by RippleNet [30]; and (2) Last-FM, a music benchmark dataset collected from the Last.fm online music system released by KGAT [37]; and (3) Alibaba-iFashion, a fashion outfit dataset collected from Alibaba online shopping systems released by KGIN [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 and Alibaba-iFashion we follow exactly the protocol given by KGAT and KGIN for processing and slicing the dataset in order to avoid inaccuracy.

Table 5: Statistics of MovieLens-1M ,Last-FM and Alibaba-iFashion

	Movielens-1M	Last-FM	Alibaba-iFashion
#Users	6036	23566	114737
#Items	2445	48123	30040
#Interactions	376886	1712638	1781093
#Entities	182011	106389	89196
#Keyphrases	179566	58266	59156
#Triplets	309172	464567	279155

The statistical information of these datasets is summarized in Table 5. 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. We use the all-ranking strategy to evaluate the recommendations, which is to rank all the items, excluding the training set.

B.2 Baselines

For a comprehensive evaluation, we conduct a thorough comparison of RISC+WER with a diverse set of baselines derived from different research areas.

Knowledge Graph Recommender Methods.

- **KGAT** [37]: It introduces the concept of collaborative knowledge graph to apply attentive aggregation on the joint user-item-entity graph.
- **KGIN** [38]: It models user intents for relations and employs relational path-aware aggregation to capture rich information from the composite knowledge graph.
- **DiffKG** [10]: It introduces the graph diffusion model, enriches the knowledge graph, and enhances the effectiveness of data augmentation.

Critiquing Methods. For critiquing models, we present the methodology used to critique and how we reproduce them in the KG recommendation model.

- **CE-VAE** [17]: It 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 embedding aggregation.
- **BK-VAE** [47]: It employs a Bayesian algorithm to update user embeddings by modifying the corresponding keyphrases. In this

research, we update the user embeddings by directly constructing a BPR loss function with the user embeddings and keyphrase embeddings.

- **DCE-VAE** [22]: It refines critiquing by constructing a keyphrase tree based on BK-VAE, allowing for more accurate mining of user feedback. Since KG inherently contains a wealth of detailed knowledge, we map keyphrases to all their neighbors to enhance their precision.
- **BCIE** [28]: 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.

Incremental Learning Methods. The datasets for these methods are directly using user critiquing as a continuous data stream.

- **iCaRL** [20]: It selects an equal number of old training samples that are closest to the feature mean of each class.
- **lwcKD** [40]: it designs contrastive learning with knowledge distillation mechanisms to preserve key model knowledge that was learned from historical data.
- **INFER** [54]: It identifies the most valuable replay examples by observing the impact of small perturbations on model predictions.

B.3 Supplemental Experimental Results

Figure 7 present the multi-step critiquing's performance of HR@5 score.