Journal of Intelligent Computing and Networking
https://www.ffspub.com/index.php/jicn/index
ISSN 3079-9228 (print)
E-mail: jicn.office@ffspub.com

Article

HRL-based Multi-Graph Fusion Framework for Sequential Recommendation

Yuandong WANG, Ding AI, Binghao ZHAN, Shangguang WANG, Pengfei WANG †
School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
†E-mail: wangpengfei@software.ict.ac.cn

Received: June 14, 2025 / Revised: June 25, 2025 / Accepted: June 27, 2025 / Published online: July 18, 2025

Abstract: In sequential recommendation, both user historical behaviors and product attribute relationships represent crucial heterogeneous information sources. Fully leveraging this diverse information is vital for enhancing recommendation performance. While recent studies demonstrate that integrating these sources can effectively boost performance, existing methods often fail to adequately address their inherent disparities. This oversight leads to semantic conflicts during fusion, diminishing recommendation accuracy and interpretability. To address this, we propose a **H**ierarchical reinforcement learning-based **Mu**lti-Graph **F**usion framework(**HuF** for short) for adaptive heterogeneous information fusion. Specifically, we first model temporal properties and product attribute relationships as graphs, and transform the fusion task into an interactive task for intelligent agents. Next, we devise a three-level agent hierarchy: low-level agents first explore paths within their respective graphs; when path explorations intertwine, the next level (middle-level agent) determines the fusion approach and evaluates information enhancement needs; subsequently, the high-level agent finally selects the most appropriate low-level agent's decision contextually. To tackle sparse coupling during learning, we introduce a rapid strategy: each low-level agent first generates paths; then, only paths exhibiting coupling are retained for subsequent high-level fusion decisions to ensure learning efficiency. We compare our model with 8 methods on three real datasets, demonstrating its effectiveness. The relevant code can be found at https://anonymous.4open.science/r/HUF/.

Keywords: Multi-graph; information fusion; hierarchical reinforcement learning; sequential recommendation; knowledge graph

1 Introduction

Sequential recommendation, based on users' historical behaviors to anticipate future decisions, stands as a highly pivotal task within the recommendation landscape. In this context, the user's temporal interaction patterns [1–4] and the inherent attribute relationships among products [5–9] manifest as two profoundly critical forms of heterogeneous information: On one hand, delving into users' behavioral sequences permits the revelation of temporal patterns, thus aiding in comprehending the evolution and trends in user behavior; on the other hand, modeling the attribute relationships among products into a knowledge graph encapsulates a wealth of semantics, substantially enhancing the understanding of user behavior and preferences.

Thus, given the ability of temporal patterns to unveil behavioral regularities and the potential of product attribute relationships to delve into inter-item associations, integrating these multifaceted pieces of information emerges as a potent approach. This integration empowers recommendation systems to better grasp user needs and provide recommendations aligned with their interests and temporal characteristics[10–12]. Undoubtedly, this represents an exceedingly effective methodology. Recent research has extensively corroborated that the significant enhancement of recommendation system performance can be achieved by effectively integrating users' temporal interaction behaviors and the inherent attribute relationships among products [13–18]. However, these methods often overlook the inconsistency among heterogeneous information, leading to semantic conflicts and mismatches during the integration process. Consequently, this diminishes the accuracy and interpretability of the recommendations.[19, 20]

† Corresponding author: Pengfei Wang
* Academic Editor: Chunxiao Jiang

Taking the association between beer and diapers as an example, although there exists a clear temporal pattern between them, the knowledge attributes between the products may not substantially assist the model in understanding this temporal pattern. Traditional approaches often involve directly incorporating knowledge information into sequential models.[21, 22] While these methods enhance the model's performance, they fall short in effectively handling the conflict between these two forms of heterogeneous information, ultimately reducing the accuracy and interpretability of the recommendations.[23–25]

In this research, we accord equal consideration to the impact of both the user's temporal interaction behaviors and the product attribute relationships on the recommendation model's performance. To proficiently merge this diverse information[26–28], we introduce a Multi-Graph network fusion framework based on hierarchical reinforcement learning. To be specific, we depict the temporal aspects of user shopping behavior and the interconnection among product attributes through graph modeling, thus converting the conventional fusion task within temporal scenarios into an interactive endeavor for intelligent agents. We formulate a three-tiered intelligent agent framework to govern the interaction and recommendation decisions between the two graphs. The low-level agent autonomously navigates paths within their respective graphs. Whenever paths explored by different agents intertwine, the medium-level agent determines the approach for fusion between them to gauge the necessity for augmenting the information. Lastly, the high-level agent selects the decision from the lower-level agents deemed suitable for the current context. To mitigate the sparsity issue encountered during the learning process due to the infrequent coupling of path exploration[29–31], we have devised an expedited learning strategy. Initially, each lower-level agent generates an array of paths. Subsequently, we retain exclusively those paths where the node coupling is observed, utilizing them for fusion decisions within the medium- and higher-level agent, thereby ensuring an effective learning trajectory for the model. We conducted a comparative analysis encompassing eight distinct methodologies on three real-world datasets, clearly showcasing the efficacy of our model. The associated code has been deposited at https://anonymous.4open.science/r/HuF/ for further in-depth exploration. The main contributions of this paper can be summarized as follows:

- We employ a Multi-Graph methodology to represent the diverse information present in temporal scenarios while maintaining their distinctiveness. Expanding upon this concept, we design a Hierarchical reinforcement learning framework to achieve fair utilization of two types of information.
- We construct a three-layered agent architecture that flexibly integrates and communicates during the exploration processes of agents on different types of information. Additionally, we have implemented a pathway selection method to ensure rapid learning of the model.
- We conduct a comparative analysis of our model with eight distinct methodologies across three real-world datasets. The results from these experiments conclusively showcased the model's superiority over a spectrum of algorithms, encompassing sequential models, recommendation models based on knowledge graphs, and hybrid models.

2 Methods

In this section, we present an intricate exposition of the Hierarchical Reinforcement Learning-based Multi-Graph Fusion framework (HUF for short) that we have introduced. The comprehensive architecture of HUF is illustrated in Figure 1. Our approach adeptly harnesses both sequential information and knowledge graph (KG) data for optimizing recommendations. Moving forward, we initiate with a comprehensive delineation of the model, elucidating the tripartite agent structure embedded within the framework.

2.1 Notations and Preliminary

Notations. Our model specifically addresses two distinct types of heterogeneous information. Regarding sequential information, we define $\mathscr{U}$ as the user set and $\mathscr{I}$ as the item set; for each user $u \in \mathscr{U}$, the interaction sequence is represented as $i^u_{1:n} = i^u_1 \to i^u_2 \to \cdots \to i^u_n$, where i^u_t denotes the item interacted by user u at the t-th time step, and n represents the sequence length. We construct a directed graph based on sequential product relationships by establishing directed links from preceding products to subsequent products. This graph, representing the practical information, is denoted as $\mathscr{G}^u_{seq}$. In terms of KG information, a knowledge graph $\mathscr{G}$ that includes an entity set $\mathscr{E}$ and a relation set $\mathscr{R}$ is defined as $\mathscr{G}_{kg} = \{(e,r,e^{'})|e,e^{'} \in \mathscr{E}, r \in \mathscr{R}\}$. Here, each triplet $(e,r,e^{'})$ denotes a fact that the head entity e is connected to the tail entity $e^{'}$ through the relation r .

Task Definition. Building upon these notations, our recommendation task aims to predict the next item interaction for user u given both sequential graph $\mathscr{G}^u_{seq}$ and knowledge graph $\mathscr{G}_{kg}$. [32]As demonstrated in [4, 33], this formulation parallels recommendation tasks in basket- or session-based settings. For conceptual clarity and simplicity, we exclusively focus on the next-session recommendation scenario throughout our model description.

Hierarchical Markov Decision Process. We first briefly introduce HMDP. [34–36] We augment HMDP with a set of goals $\mathscr{G}$, where each goal represents a state or set of states that an agent should learn to achieve. Formally, a HMDP is defined as a tuple $\mathscr{U} = (\mathscr{S}, \mathscr{G}, \mathscr{A}, \mathscr{T}, \mathscr{R}, \gamma)$ where $\mathscr{S}$ denotes the state set, $\mathscr{G}$ the goal set, $\mathscr{A}$ the action set, $\mathscr{T}$ the transition probability function (specifically, $\mathscr{T}(s,a,s')$ gives the probability of transitioning to state s' when executing action a in state s), $\mathscr{R}$ the reward function, and $\gamma \in [0,1)$ the discount factor. At each episode's inception, a goal $g \in \mathscr{G}$ is selected and remains fixed throughout the episode. The solution involves finding an optimal control policy $\pi : \mathscr{S} \times \mathscr{G} \to \mathscr{A}$ that maximizes the value function $v_\pi(s,g) = \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} \gamma^n R_{t+n+1} \,\middle|\, s_t = s, g_t = g \right]$ for initial state s and goal g.

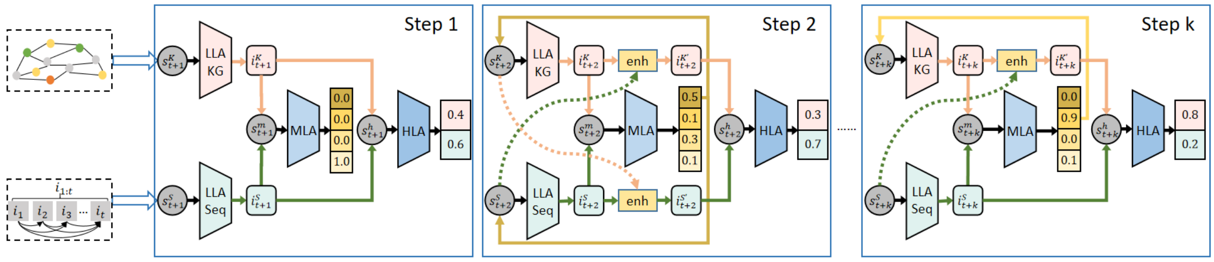

Figure 1: The overall architecture of Hierarchical Reinforcement Learning-based Multi-Graph Fusion framework (HUF for short). The Low-level Agents(LLA) provide item recommendations based on a corresponding single information source; the mid-level agent(MLA) is used to determine whether there is information exchange between low-level agents for decision refinement, and the high-level agent(HLA) determines which low-level agent's decision should be utilized in the current context.

2.2 Model Overview

Figure 1 shows the overall architecture of HUF , primarily composed of Decisions of Low-level Agents, Enhancement on Medium-level Agent, and Item Selector on High-level Agent: The Low-level Agents provide item recommendations based on a corresponding single information source; the mid-level agent is used to determine whether there is information exchange between low-level agents for decision refinement, and the high-level agent determines which low-level agent's decision should be utilized in the current context.

In our model, we have a total of three layers of agents. In this chapter, we will first elaborate on the actions and states of the three layers of agents:

- **states** and **actions** for Low-level agents: HUF maintains two distinct low-level agents for exploration in the two graphs. At the t-th time step, the states of these two agents are defined as $s_l^{seq}(t)$ and $s_l^{kg}(t)$, and their corresponding actions are denoted as $a_l^{seq}(t)$ and $a_l^{kg}(t)$ respectively.A detailed introduction will be given in Section 2.3.
- **states** and **actions** for Mid-level action: The mid-level agent bases its action decision on the current state $s_m(t)$, resulting in the action denoted as $a_m(t)$ to determine whether to engage in information exchange. $a_m(t)$ represents the decision probabilities for performing four different actions.A detailed introduction will be given in Section 2.4.
- **states** and **actions** for High-level agent: Given the current state $s_h(t)$of the high-level agent, it outputs an action $a_h(t)$ that determines which decision from the low-level agents to retain.A detailed introduction will be given in Section 2.5.

2.3 Decisions of Low-level Agents

HUF manages two low-level agents to explore items on two graphs separately. Below, we will introduce each of these agents separately.

2.3.1 Low-level Agent for Sequential Property

On $\mathcal{G}^u_{seq}$, Baby selects items based on the current state $s_l^{seq}(t)$. Specifically, we construct $s_l^{seq}(t)$ using a GRU [37] as the sequential state encoder:

$$s_l^{\mathrm{seq}}(t) = \mathrm{GRU}\left[s_l^{\mathrm{seq}}(t-1); i_{a_l^{\mathrm{seq}}(t)}\right] \tag{1}$$

where $\mathbf{i}_{a_l^{seq}(t)}$ denotes the embedding vector for the t-th item in $\mathcal{G}^u_{seq}$. The initial state is defined as $\mathbf{s}_l^{seq}(0) = \mathbf{GRU}[\mathbf{i}_{1:n}]$. Subsequently, at each timestep t, the agent selects an action $a_l^{seq}(t)$ based on the current state $s_l^{seq}(t)$, which corresponds to recommending the next item i_{t+1} from the item set $\mathcal{I}$. This action selection is governed by a policy $\pi(s_l^{seq}(t))$ that maps the state $s_l^{seq}(t)$ to a probability distribution over all possible items; in this work, we implement this policy using a softmax function to compute item selection probabilities:

$$\pi\left(a_l^{\mathrm{seq}}(t) \mid s_l^{\mathrm{seq}}(t)\right) = \frac{\exp\left\{i_{a_l^{\mathrm{seq}}(t)} \cdot s_l^{\mathrm{seq}}(t)\right\}}{\sum\limits_{i \in \mathcal{I}} \exp\left\{i \cdot s_l^{\mathrm{seq}}(t)\right\}} \tag{2}$$

2.3.2 Low-level Agent for KG Property

The Knowledge Graph (KG) model formalizes the KGRE-Rec problem as a Markov Decision Process (MDP) [38], defining the state $\mathbf{s}_l^{kg}(t)$ as a tuple (u, e_t, h_t) where $u \in \mathcal{U}$ represents the starting user entity, e_t denotes the entity reached by the agent at step t, and h_t encapsulates the history prior to step t—with the k-step history specifically constructed as the combination of all entities and relations from the preceding k steps. Given an initial user u, the starting state is defined as $\mathbf{s}_l^{kg}(0) = (\mathbf{u}, \mathbf{u}, \emptyset)$. In this paper, the KG model has $k = 1$. Therefore, the state at step t is:

$$s_l^{\mathrm{kg}}(t) = u \oplus i_{a_l^{\mathrm{kg}}(t)} \oplus i_{a_l^{\mathrm{kg}}(t-1)} \oplus r_t \tag{3}$$

where "$\oplus$" is the vector concatenation operator, r_t is the relation between entities $i_{a_l^{kg}(t-1)}$ and $i_{a_l^{kg}(t)}$. The complete action space A_t of state $s_l^{kg}(t)$ is defined as all possible outgoing edges of entity $i_{a_l^{kg}(t)}$ excluding history entities and relations. KG model introduces a user-conditional action pruning strategy to sample action. Specifically, the scoring function $f((r,i)|u)$ maps any edge (r,i) to a real-valued score conditioned on user u. Then, the user-conditional pruned action space of state $s_l^{kg}(t)$, denoted by $A_t(u)$, is defined as:

$$A_t(u) = \left\{i \mid \mathrm{rank}\big(f((r,i) \mid u)\big) \leq \alpha,\ (r,i) \in A_t\right\} \tag{4}$$

where α is a pre-defined integer that upper-bounds the size of the action space. KG model's action behavior policy is defined as:

$$\pi\left(a_l^{\mathrm{kg}}(t) \mid s_l^{\mathrm{kg}}(t)\right)=\frac{\exp\left\{i_{a_l^{\mathrm{kg}}(t)} \cdot s_l^{\mathrm{kg}}(t)\right\}}{\sum\limits_{i \in A_t(u)} \exp\left\{i \cdot s_l^{\mathrm{kg}}(t)\right\}} \tag{5}$$

2.4 Enhancement on Middle-level Agent

MLA is an Information Enhancement selector. It is used to determine whether the recommended items in the LLA undergo Information Enhancement. If not, the recommended items in LLA are directly uploaded to HLA as the recommended items in MLA. If yes, it's necessary to enhance the recommended items in LLA with additional information from other information sources, and the enhanced items are uploaded to HLA as the recommended items in MLA.

In LLA, reinforcement learning is performed independently between different information sources. Each agent independently recommends the best items for the user from its own graph. These recommended items reflect the user's preferences for different information sources from a side perspective. In some cases, different information sources may recommend the same item, indicating a crossover between information sources in terms of this user's preferences. We believe that by using the historical recommendation information, we can enhance the recommended items. Through experiments (section 4.4.1), we demonstrate that this method helps improve the performance of the recommendation algorithm.

We construct the state $s_m(t)$ of MLA using three vectors. The user vector and the vectors of items recommended by LLA through sequential information and KG information:

$$s_m(t)=u \oplus i_{a_l^{\mathrm{kg}}(t)} \oplus i_{a_l^{\mathrm{seq}}(t)} \tag{6}$$

MLA is used to determine whether the items recommended by LLA should undergo Information Enhancement. In this paper, we use both sequence and KG information, and the action space size of MLA is 4. The MLA generates actions according to the state embeddings $s_m(t)$. Specifically, we feed $s_m(t)$ into *MLP* and a fully connected layer to generate the t-th MLA action embedding $a_m(t)$ as:

$$a_m(t)=\operatorname{MLP}\left(W_m \cdot s_m(t)+b_m\right) \tag{7}$$

Where W_m and b_m are the parameters. $a_m(t)$ can take four different values, which are only enhance $i_l^{seq}(t)$ by using KG information, only enhance $i_l^{kg}(t)$ by using sequential information, enhance both, or not enhance either. Regardless of whether the items are enhanced, they are eventually uploaded as $i_{a_m^{kg}(t)}$ and $i_{a_m^{seq}(t)}$ to HLA.

The following provides a detailed explanation of how to perform Information Enhancement. Since different LLA agents are heterogeneous agents, the enhancement mechanism for items varies between different information sources. First is enhancing sequential information by using KG information. As shown in Huang et al.[4], KG information contributes to improving the performance of sequential recommendation algorithms. However, previous methods primarily focused on enhancing item or user representations with KG information to model users' short-term behavior[39]. In contrast, we achieve information enhancement by incorporating KG historical recommended item sequences into the sequential information model. For example, when the KG recommended item, denoted as $i_{a_l^{kg}(t)}$, matches the sequence recommended item, $i_{a_l^{seq}(t)}$:

$$s_l^{\mathrm{seq}}(t)=\operatorname{GRU}\left[s_l^{\mathrm{seq}}(t) ; s_l^{\mathrm{kg}}(t)\right] \tag{8}$$

At the same time, because the KG model only focuses on the recommended items from the last step in the history:

$$s_l^{\mathrm{kg}}(t)=u \oplus i_{a_l^{\mathrm{kg}}(t)} \oplus i_{a_l^{\mathrm{seq}}(t-1)} \oplus r_t \tag{9}$$

When $s_l(t)$ in LLA changes, the action $a_l(t)$ will also naturally differ after going through Eq. (2) and (5) The final recommended items, as a result, will be uploaded to HLA as $i_{a_m^{seq}(t)}$ and $i_{a_m^{kg}(t)}$.

2.5 Item Selector on High-level Agent

HLA is an Item Selector. It is used to determine the final k items for recommendation from the $k \times x$ items recommended by MLA.

Similar to $s_m(t)$ in MLA, we construct the t-th time step state $s_h(t)$ of HLA by using three vectors. The user vector and the items vector recommended by MLA using sequential information and KG information:

$$s_h(t)=u \oplus i_{a_m^{\mathrm{kg}}(t)} \oplus i_{a_m^{\mathrm{seq}}(t)} \tag{10}$$

HLA is used to determine the final recommended items from $i_{a_m^{seq}(t)}$ and $i_{a_m^{kg}(t)}$. In this paper, we use both sequence and KG information, so the action space size of HLA is 2. The policy network $\pi(a_h(t)|s_h(t),A_h)$ takes as input the state vector $s_h(t)$ and binarized vector A_h and emits the probability of each action.

Defining an appropriate reward function is crucial for reinforcement learning algorithms, particularly in recommendation systems where performance is typically evaluated through exact item ID matching. Drawing inspiration from machine translation evaluation methods [40], we adapt the *BLEU* metric for recommendation tasks. Formally, given an actual user interaction subsequence $i_{t:t+k}$ and the corresponding recommended subsequence $i^h_{t:t+k}$, we formulate the reward function as follows:

$$R_h=\exp\left(\frac{1}{N} \sum_{n=1}^{N} \log prec_n\right) \tag{11}$$

where $prec_n$ is the modified precision. This reward function advocates the recommendation algorithm to generate more consistent m-grams from the actual sequence.

2.6 Learning and Discussion

Based on the MDP formulation, our goal is to learn a stochastic policy π that maximizes the expected cumulative reward for any user:

$$J(\theta) = \mathbb{E}_\pi \left[\sum_{t=1}^{k} \gamma^t R_t \mid s_h(t) \right] \tag{12}$$

We solve the problem through designing a policy network and a value network that share the same feature layers. The model parameters for both networks are denoted as $\theta = \{W_1, W_2, W_p, W_v\}$

There is a challenge in the policy learning process, as the product sequences generated by various heterogeneous agents through LLA, denoted as $i^S_{t:t+k}$ and $i^K_{t:t+k}$, may not necessarily contain the same products. This is due to the differences in action spaces between the heterogeneous agents. In this paper, the action space of the sequential model includes all items, while the action space of the KG model is generated through sampling. This difference can result in some users having no repeated item sequences in LLA training. To improve this issue, We have developed an expedited selection strategy that entails filtering the paths autonomously generated by lower-level intelligent agents. Specifically, we retain only those paths that demonstrate coupling. Subsequently, these carefully selected paths are employed to optimize the model, guaranteeing an accelerated learning process.

3 Results

In this section, we extensively evaluate the performance of the HuF method on real datasets. We begin by introducing our benchmarks and the corresponding experimental settings. Then, we quantitatively compare the effectiveness of our model with other state-of-the-art methods and then conduct ablation studies to demonstrate how variations in the model affect our experimental results.

3.1 Data Description

All experiments were conducted on the Amazon e-commerce datasets collection[41, 42], which includes product reviews and meta information from Amazon.com. These datasets encompass three categories: Clothing, Cell Phones and Beauty. For all datasets, we removed users and items with fewer than 5 interaction records. In our work, we required access to the knowledge graph information for each item in the dataset. Each dataset is treated as a separate benchmark, forming a knowledge graph containing 5 entity types and 7 relations. Descriptions and statistics for each entity and relation can be found in Table 1. We utilized the same data preprocessing method by following[43–45]. Regarding data splitting rules, for each user, we sorted their records based on timestamps to form interaction sequences. Based on the sorted sequences, we used the first 80% of purchase records as training data and the remaining 20% as test data.

3.2 Experimental Setup

Baselines. We adopt three types of baselines for comparison, including sequential-based models, knowledge-based models, and hybrid models.

- *GRU4Rec*[37] is a session-based recommendation, which utilizes GRU to capture users' long-term sequential behaviors.
- *CORE*[46] proposes a robust distance measuring method to prevent overfitting of embeddings in the consistent representation space to unify the representation space for both the encoding and decoding processes for recommendation.
- *CKE*[5] is a contemporary neural recommendation system founded on a joint modeling approach that integrates matrix factorization with heterogeneous data modalities—encompassing textual content, visual information, and structural knowledge bases—to generate top-N recommendation results.
- *RippleNet*[47] is an embedding-based method that models users' potential interests along links in the knowledge graph for recommendation.
- *KGAT*[48] explores the high-order connectivity with semantic relations in collaborative knowledge graph for knowledge-aware recommendation.
- *MCCLK*[49] hence performs contrastive learning across three views on both local and global levels, mining comprehensive graph feature and structure information in a self-supervised manner for recommendations.
- *KERL*[50] adopts an RL model to make recommendations over KG. In the MDP modeled by KERL, the environment contains the information of interaction data and KG, which are useful for the sequential recommendation.
- *PGPR*[43] performs recommendations and explanations by providing actual paths with the REINFORCE algorithm over the KG.

Evaluation Metrics. For each a user, a method will produce a top-N recommendation list for evaluation. Following, we employ four representative top-N recommendation measures: Normalized Discounted Cumulative Gain (NDCG), Recall, Hit Ratio (HR) and Precision (Prec.). k is set to 5 in our experiments.

Parameter Settings. In our model, sequential recommendation and KG-based recommendation are respectively performed using the $\mathrm{KERL_h}$ and PGPR models. For the $\mathrm{KERL_h}$ model, the hidden layer size of GRU and item embeddings are set to 100, and the KG embeddings using TransE[51] have a size of 50. Regarding hyperparameters, the discount factor γ for the next item recommendation and the next session recommendation tasks is set to 0.9, and the window size k is set to 3. For the PGPR model, all model parameters remain the same as in the original paper[43]. For our model, we set the batch size to 64, the learning rate to 0.001, the discount factor γ to 0.99, and the weight for the entropy loss to 0.001. Since both the $\mathrm{KERL_h}$ and PGPR models have embedding sizes of 100, the size of the state vector $s_t = (u, i_s, i_k)$ is 300. For the policy/value network, $W_m \in \mathbb{R}^{300\times4}$, $W_1 \in \mathbb{R}^{300\times512}$, $W_2 \in \mathbb{R}^{512\times256}$, $W_p \in \mathbb{R}^{256\times2}$ and $W_v \in \mathbb{R}^{256\times1}$.

3.3 Performance Comparison

In this experiment, we quantitatively evaluated the performance of our model on recommendation tasks and compared it with other baselines on all three Amazon datasets. We used the default settings described in the previous section.

Table 1: Statistics of datasets for experiments (a.v.l = average sequence length; a.v.e = average entity edges).

Dataset	#interactions	#users	#items	#a.v.l	#relations	#entities	#a.v.e
Beauty	198,502	22,363	12,101	8.88	8,200,842	224,074	36.60
Cell	194,439	27,879	10,429	6.97	6,419,242	163,249	39.32
Clothing	278,677	39,387	23,033	7.08	10,714,118	425,528	25.18

Overall, Our HuF method consistently outperforms all other baselines on all datasets in terms of NDCG, Hit Ratio, Recall, and Precision. We present the comparison results in Table 2. From this table, we can observe that:

Looking at the overall trend, all models follow a similar pattern: For each model individually, their performance metrics are highest on the Cell dataset, lowest on the Clothing dataset, and somewhere in between on the Beauty dataset.

For sequential models, their overall performance metrics are lower than KG models and hybrid models. They only achieve metrics similar to or slightly better than KG models in the Cell dataset, which is the best-performing dataset. In the other two datasets, sequential models perform the worst.

For KG models, their performance metrics are relatively stable overall. Except for the Cell dataset where the metrics are lower than some individual sequential models and the Clothing dataset where the metrics are higher than some individual hybrid models, KG models generally have metrics higher than sequential models and lower than hybrid models overall.

For hybrid models, except for KERL having lower metrics than some individual KG models on the Clothing dataset, their overall performance metrics are the highest among the baselines on all three datasets. However, different hybrid models focus on different types of information. The KERL model emphasizes sequential information, which is why it performs better on the Cell dataset. On the other hand, the PGPR model emphasizes KG information, leading to higher metrics on the Beauty and Clothing datasets. The MCCLK model falls in between, with slightly better Hit Ratio metrics on the Cell dataset compared to the KERL model.

Finally, our proposed HuF method performs the best among all methods on the three datasets. While other hybrid models[43, 49, 50] also utilize some KG information and some sequential information, they tend to lose some information during the fusion process. What sets HuF apart is its ability to retain the full information from both sources and make recommendations through hierarchical reinforcement learning. With this meaningful extension, our model can better leverage KG and sequential information for recommendations.

The performance of the HuF model varies across different datasets. Based on the data from Table 1, we can observe that the Clothing dataset has a much larger number of items (#items=23033) compared to the other datasets, almost twice the size. This results in relatively sparse information overall. Additionally, in the Clothing dataset, the value of #a.v.l=7.08, which is the lowest among the datasets. This indicates that its sequential information is significantly weaker compared to the KG information. As a result, our model shows a lower improvement on the Clothing dataset. On the contrary, in the Cell dataset, although the sequential information is stronger than KG information, it is in a better condition compared to the Clothing dataset. This may be due to the characteristics of cellphones as products. Our model shows a higher improvement in this case, but there is still room for further improvement. The balance between sequential information and KG information in the Beauty dataset prevents our model from leaning too much towards one aspect. By fusing information, we ultimately achieved the highest performance improvement on the Beauty dataset.

Table 2: Top-5 Recommendation Performance: Baselines vs. Our Model.The best performance in each column is highlighted in bold, with the best baseline performance shown in red.

Dataset	Model	NDCG	Recall	HR	Prec.
Beauty	GRU4Rec	1.093	1.822	2.348	0.485
	CORE	1.562	2.235	2.549	0.565
	RippleNet	1.171	1.988	2.352	0.382
	CKE	1.394	2.215	2.938	0.499
	KGAT	1.593	2.595	2.758	0.529
	MCCLK	1.844	2.839	3.161	0.643
	KERL	1.784	2.616	3.37	0.648
	PGPR	1.965	3.063	3.43	0.698
	HuF(Ours)	**2.16**	**3.215**	**3.832**	**0.808**
	Improv.(%)	**9.9**	**5.0**	**11.7**	**15.8**
Cell	GRU4Rec	1.764	3.02	3.375	0.69
	CORE	2.048	3.286	3.567	0.707
	RippleNet	1.726	2.61	2.321	0.423
	CKE	1.847	2.871	3.039	0.593
	KGAT	1.97	2.818	3.293	0.635
	MCCLK	2.155	3.359	3.771	0.762
	KERL	2.242	3.532	3.745	0.814
	PGPR	2.067	3.201	3.508	0.708
	HuF(Ours)	**2.345**	**3.813**	**3.978**	**0.859**
	Improv.(%)	**4.6**	**8.0**	**5.5**	**5.5**
Clothing	GRU4Rec	0.218	0.381	0.46	0.092
	CORE	0.412	0.875	0.951	0.178
	RippleNet	0.602	0.731	0.978	0.286
	CKE	0.681	0.897	1.186	0.273
	KGAT	0.847	0.976	1.167	0.305
	MCCLK	1.016	1.182	1.725	0.343
	KERL	0.7	1.079	1.777	0.363
	PGPR	1.112	1.698	1.881	0.378
	HuF(Ours)	**1.135**	**1.756**	**1.94**	**0.396**
	Improv.(%)	**2.1**	**3.4**	**3.1**	**4.8**

3.4 Ablation Study

HuF has made several important extensions to integrate KG and sequential information into RL for recommendations. In this section, we conduct ablation experiments to analyze their impact.

3.4.1 Influence of Data enhancement methods

In this section, we analyzed the impact of different Information Enhancement methods on the experimental results. Recalling that we perform Information Enhancement when sequential and KG information recommend the same item and after completing the first round of Information Enhancement, the same information source is no longer used for further Information Enhancement. Therefore, we compare three variants to consider the impact of each component on recommendations, including:

- *HuF-enh*: No Information Enhancement
- *HuF-time*: Remove the restriction of Information Enhancement only once; it can be performed multiple times, up to k times.
- *HuF-item*: Remove the restriction of recommend the same item; Information Enhancement can be performed regardless of the recommended items by sequential and KG information.

The results for HuF and its three variants on the Beauty dataset are shown in Table 3. We observe that the HuF-item model performs the worst in all evaluation metrics. This could be because information enhancement for non-repeated recommended items introduces too much noise. The HuF-time model performs slightly better in all evaluation metrics, but it is still worse than the HuF-enh model. This is likely due to the excessive times of Information Enhancement, which severely impacted the original information and resulted in overfitting issues. Finally, by constraining the conditions and the number of times on Information Enhancement, the complete HuF model outperforms all its three variants on all evaluation metrics.

Table 3: Performance comparison of HuF-enh, HuF-time, HuF-item and HuF over Beauty dataset. Best performance is in bold font.

Metric	**NDCG@5**	**Recall@5**	**HR@5**	**Precision@5**
HuF-enh	2.145	3.204	3.716	0.8
HuF-time	2.14	3.166	3.761	0.795
HuF-item	2.107	3.117	3.707	0.77
HuF	**2.16**	**3.215**	**3.832**	**0.808**

3.4.2 Influence of Hierarchical Module Low&High

In this section, we analyze the impact of HLA and MLA on the experimental results. HLA selects from which information source to obtain the recommended items. MLA selects whether to perform information enhancement and recommends the exact items. Therefore, we compare three variants and one baseline to consider the impact of each component on recommendations, including:

- *$KERL_h$*: Variant of KERL model uses only sequence information.
- *$KERL_h$+enh*: $KERL_h$ model after MLA.
- *PGPR*: PGPR model, same as the baselines in Table 2.
- *PGPR+enh*: PGPR model after MLA.

HuF and its three variants along with a baseline on the Beauty dataset are presented in Table 4. We observe that the $KERL_h$+enh model exhibits improvements in all evaluation metrics compared to $KERL_h$. On the other hand, the PGPR+enh model shows a decrease in all evaluation metrics compared to the PGPR model. This is because when both models integrate information from the other, it affects the balance of information, leading to information preference transfer. To obtain better results, filtering through HLA is necessary. The Table 4 shows that the HuF model outperforms its three variants and the baseline in all evaluation metrics.

Table 4: Performance comparison of $KERL_h$, $KERL_h$+enh, PGPR,PGPR+enh and HuF over Beauty dataset. Best performance is in bold font.

Dataset	Beauty			
Metric	NDCG@5	Recall@5	HR@5	Precision@5
$KERL_h$	1.085	1.802	2.245	0.469
$KERL_h$+enh	1.097	1.826	2.352	0.486
PGPR	1.965	3.063	3.43	0.698
PGPR+enh	1.953	3.045	3.273	0.66
HuF	**2.16**	**3.215**	**3.832**	**0.808**

3.4.3 Influence of Sequential or Knowledge Graph Parameters

In this section, we analyze the impact of the parameters of the $KERL_h$ model and the PGPR model on the experimental results. The most significant parameter in the $KERL_h$ model that affects the experimental results is the subsequence length, denoted as k. The most crucial parameter in the PGPR model that affects the experimental results is the sampling sizes at each level. Therefore, we compare five variants and one baseline to consider the impact of each parameter on the recommendations. The results are shown in Table 5

Table 5: Performance comparison of five variants, one baseline and HuF over Beauty dataset. Best performance is in bold font.

Dataset	Beauty			
Metric	NDCG@5	Recall@5	HR@5	Precision@5
$KERL_h$	1.085	1.802	2.245	0.469
$KERL_h$(k=5)	1.013	1.702	2.206	0.458
HuF(k=5)	2.025	3.103	3.695	0.744
PGPR	1.965	3.063	3.43	0.698
PGPR[10,5,1]	1.774	2.845	3.083	0.647
HuF[10,5,1]	1.979	3.075	3.537	0.742
HuF	**2.16**	**3.215**	**3.832**	**0.808**

The default parameter for the $KERL_h$ model is k=3. We observe that when k=5, all evaluation metrics for the $KERL_h$ model decline. This could be due to the longer search window possibly containing noisy information, which can affect the recommendation performance, in line with the findings in the original paper [50]. In the PGPR model, the default parameters for ”sampling sizes at each level” are set as [25,5,1]. When the sampling sizes at each level in the PGPR model are set to [10,5,1], all the evaluation metrics show a decrease. This might be due to the smaller sampling sizes, which result in fewer paths being sampled and could lead to the loss of items with higher ratings. When the evaluation metrics for individual models decrease, the evaluation metrics for the HuF model also decrease in tandem. From the Table 5, we observe that the decrease in evaluation metrics for the $KERL_h$ and PGPR models is roughly proportional to the decrease in

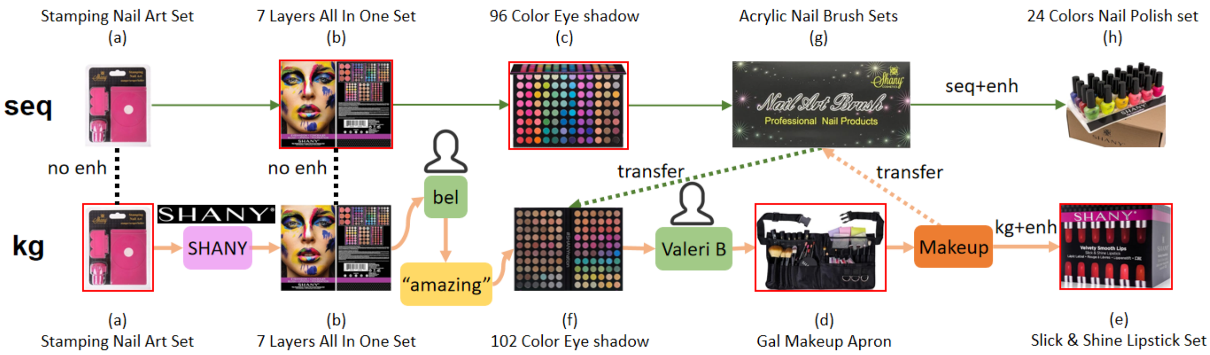

Figure 2: Real cases of recommendation reasoning paths. The items in the redlining are the final recommended items. The dashed arrows represent the recommended items that transition after information enhancement.

evaluation metrics for the HuF model. This result suggests that, for the Beauty dataset, both sequence information and KG information have similar levels of importance.

3.5 Case Study

In our model, a significant novelty points is that we independently use sequential information and KG information to improve recommendation. Prior experimental results have demonstrated the effectiveness of our approach in enhancing recommendation performance; to elucidate the underlying mechanisms contributing to its utility, we conduct a qualitative analysis through a case study on the Beauty dataset, as presented in Figure 2.

Specially, we present a snapshot of the purchasing sequence for a sample user. The interaction sequence is time-ordered, consisting of five items. The first two items are ”Stamping Nail Art Set” and ”7 Layers In One Makeup Set.” The next three items are ”96 Color Eye Shadow Palette,” ”Makeup Apron,” and ”Lipstick Set.” Based on this snapshot, the user might be a professional makeup artist and needs a wide range of beauties.

To illustrate the usefulness of HuF model, we will compare it with the $\mathrm{KERL_h}$ and PGPR models. From the results, both $\mathrm{KERL_h}$ and PGPR models seem to get ”stuck” in the partial user preferences. For example, $\mathrm{KERL_h}$ model might assume that the user prefers buying various makeup sets, and PGPR model recommends an eyeshadow palette with more colors. HuF is able to successfully capture the user's preference drift by selecting items (recommending through both sequential and KG information simultaneously).

As shown in Figure 2, the sequential model in LLA recommended items (a,b,c,g,f), while the KG model recommended items $(a,b,f,$ $d,g)$. There are four repeated recommended items: (a,b,f,g). After MLA information enhancement, the sequential model recommended items (a,b,c,g,h), while the KG model recommended items (a,b,f,d,e). Item i_h is enhanced from i_f, and i_e is enhanced from i_g. The repeated items (a,b) were not enhanced in MLA. Finally, after HLA item selection, the user was recommended items (a,b,c,d,e), which is the best result among all five item sequences.

4 Discussion

In this paper, we propose a Hierarchical reinforcement learning-based Multi-Graph Fusion framework, called HuF. This is used to ensure the fair utilization of both types of information while maintaining the uniqueness of sequential and KG information. Specifically, we have established a three-level agent architecture, which enables flexible integration and communication among agents during the exploration of different types of information. Additionally, we have also implemented path selection methods to ensure the model's rapid learning. One of the key novelties in our model is that we perform information enhancement when low-level agents recommend the same items repeatedly. The experimental results demonstrate that our model outperforms the baselines significantly on three real datasets. We also conducted a detailed analysis of the HuF model to illustrate the effectiveness of our extensions. Currently, we are focused on utilizing only sequential information and KG information. Additionally, our information enhancement method is limited to using past recommended product sequences. As future work, we will consider how to incorporate more diverse information into the framework and explore a wider range of methods for information enhancement.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions

Conceptualization, Pengfei Wang; methodology, Yuandong Wang; software, Yuandong Wang; validation, Pengfei Wang; formal analysis, Binghao Zhan; investigation, Binghao Zhan; resources, Shangguang WANG; data curation, Yuandong Wang; writing—original draft preparation, Yuandong Wang; writing—review and editing, Pengfei Wang; visualization, Ding Ai; supervision, Pengfei Wang; project administration, Binghao Zhan; All authors have read and agreed to the published version of the manuscript.

Conflict of Interest

All the authors declare that they have no conflict of interest.

Data Available

The data and materials used in this study are available upon request from the corresponding author.

References

[1] Lin, G., Gao, C., Zheng, Y., Chang, J., Niu, Y., Song, Y., Li, Z., Jin, D., Li, Y.: Dual-interest factorization-heads attention for sequential recommendation. In Proceedings of the ACM Web Conference 2023 (WWW '23), pp. 917–927 (2023)

[2] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S.: Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web (WWW '17), pp. 173–182 (2017)

[3] Donkers, T., Loepp, B., Ziegler, J.: Sequential user-based recurrent neural network recommendations. In Proceedings of the 11th ACM Conference on Recommender Systems (RecSys '17), pp. 152–160 (2017)

[4] Huang, J., Zhao, W.X., Dou, H., Wen, J., Chang, E.Y.: Improving sequential recommendation with knowledge-enhanced memory networks. In The 41st International ACM SIGIR Conference on Research Development in Information Retrieval (SIGIR '18), pp. 505–514 (2018)

[5] Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.-Y.: Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), pp. 353–362 (2016)

[6] Zou, D., Wei, W., Wang, Z., Mao, X., Zhu, F., Fang, R., Chen, D.: Improving knowledge-aware recommendation with multi-level interactive contrastive learning. In Proceedings of the 31st ACM International Conference on Information Knowledge Management (CIKM '22), pp. 2817–2826 (2022)

[7] Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., Guo, M.: Multi-task feature learning for knowledge graph enhanced recommendation. In The World Wide Web Conference (WWW '19), pp. 2000–2010 (2019)

[8] Liu, Y., Miyazaki, J., Chang, Q.: Knowledge graph enhanced multi-task learning between reviews and ratings for movie recommendation. In Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22), pp. 1882–1889 (2022)

[9] Shen, X., Zhang, Y.: A knowledge graph recommendation approach incorporating contrastive and relationship learning. IEEE Access **11**, 99628–99637 (2023)

[10] Malhotra, R., Taneja, A., Arora, A.: Temporal recommendations for discovering author interests. In 2019 11th International Conference on Communication Systems Networks (IC3), pp. 1–6 (2019)

[11] Ding, H., Liu, Q., Hu, G.: Tdtmf: A recommendation model based on user temporal interest drift and latent review topic evolution with regularization factor. Information Processing & Management **59**(5), 103037 (2022)

[12] Zhao, K., Wang, X., Zhang, Y., Zhao, L., Liu, Z., Xing, C., Xie, X.: Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 239–248 (2020)

[13] Lin, Y., Liu, Y., Lin, F., Zou, L., Wu, P., Zeng, W., Chen, H., Miao, C.: A survey on reinforcement learning for recommender systems. IEEE Transactions on Neural Networks and Learning Systems **35**(10), 13164–13184 (2023)

[14] Zhu, T., Sun, L., Chen, G.: Graph-based embedding smoothing for sequential recommendation. IEEE Transactions on Knowledge and Data Engineering **35**(1), 496–508 (2021)

[15] Liu, J., Zhuo, H.H., Jin, K., Yuan, J., Yang, Z., Yao, Z.: Sequential condition evolved interaction knowledge graph for traditional chinese medicine recommendation. arXiv preprint arXiv:2305.17866 (2023)

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

[17] Xin, X., Karatzoglou, A., Arapakis, I., Jose, J.M.: Self-supervised reinforcement learning for recommender systems. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 931–940 (2020)

[18] Antaris, S., Rafailidis, D.: Sequence adaptation via reinforcement learning in recommender systems. In Proceedings of the 15th Acm Conference on Recommender Systems, pp. 714–718 (2021)

[19] Chen, Q., Guo, Z., Li, J., Li, G.: Knowledge-enhanced multi-view graph neural networks for session-based recommendation. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 352–361 (2023)

[20] Fu, Z., Xian, Y., Gao, R., Zhao, J., Huang, Q., Ge, Y., Xu, S., Geng, S., Shah, C., Zhang, Y., *et al.*: Fairness-aware explainable recommendation over knowledge graphs. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in

Information Retrieval, pp. 69–78 (2020)

[21] Cui, Z., Yuan, Z., Wu, Y., Sun, X., Yu, K.: Intelligent recommendation for departments based on medical knowledge graph. IEEE Access **11**, 25372–25385 (2023)

[22] Geng, S., Fu, Z., Tan, J., Ge, Y., De Melo, G., Zhang, Y.: Path language modeling over knowledge graphsfor explainable recommendation. In Proceedings of the ACM Web Conference 2022, pp. 946–955 (2022)

[23] Zhang, J., Hao, B., Chen, B., Li, C., Chen, H., Sun, J.: Hierarchical reinforcement learning for course recommendation in moocs. In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 435–442 (2019)

[24] Xie, R., Zhang, S., Wang, R., Xia, F., Lin, L.: Hierarchical reinforcement learning for integrated recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 4521–4528 (2021)

[25] Wang, L., Tang, R., He, X., He, X.: Hierarchical imitation learning via subgoal representation learning for dynamic treatment recommendation. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 1081–1089 (2022)

[26] Wang, E., Ding, R., Yang, Z., Jin, H., Miao, C., Su, L., Zhang, F., Qiao, C., Wang, X.: Joint charging and relocation recommendation for e-taxi drivers via multi-agent mean field hierarchical reinforcement learning. IEEE Transactions on Mobile Computing **21**(4), 1274–1290 (2020)

[27] Du, Q., Yu, L., Li, H., Leng, Y., Ou, N.: Denoising-oriented deep hierarchical reinforcement learning for next-basket recommendation. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4093–4097 (2022). IEEE

[28] Ni, C.-C., Sum Liu, K., Torzec, N.: Layered graph embedding for entity recommendation using wikipedia in the yahoo! knowledge graph. In Companion Proceedings of the Web Conference 2020, pp. 811–818 (2020)

[29] Wu, J., Xie, Z., Yu, T., Zhao, H., Zhang, R., Li, S.: Dynamics-aware adaptation for reinforcement learning based cross-domain interactive recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 290–300 (2022)

[30] Luo, J., Hazra, K.S., Huo, W., Li, R., Mahabal, A.: Personalized style recommendation via reinforcement learning. In Companion Proceedings of the ACM Web Conference 2023, pp. 290–293 (2023)

[31] Lin, Y., Lin, F., Zeng, W., Xiahou, J., Li, L., Wu, P., Liu, Y., Miao, C.: Hierarchical reinforcement learning with dynamic recurrent mechanism for course recommendation. Knowledge-Based Systems **244**, 108546 (2022)

[32] Bing, Q., Zhu, Q., Dou, Z.: Cognition-aware knowledge graph reasoning for explainable recommendation. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pp. 402–410 (2023)

[33] Liu, Z., Tian, J., Cai, Q., Zhao, X., Gao, J., Liu, S., Chen, D., He, T., Zheng, D., Jiang, P., *et al.*: Multi-task recommendations with reinforcement learning. In Proceedings of the ACM Web Conference 2023, pp. 1273–1282 (2023)

[34] Lin, Y., Feng, S., Lin, F., Zeng, W., Liu, Y., Wu, P.: Adaptive course recommendation in moocs. Knowledge-Based Systems **224**, 107085 (2021)

[35] Duan, J., Zhang, P.-F., Qiu, R., Huang, Z.: Long short-term enhanced memory for sequential recommendation. World Wide Web **26**(2), 561–583 (2023)

[36] Wang, X., Huang, T., Wang, D., Yuan, Y., Liu, Z., He, X., Chua, T.-S.: Learning intents behind interactions with knowledge graph for recommendation. In Proceedings of the Web Conference 2021, pp. 878–887 (2021)

[37] Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)

[38] Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. The MIT Press, Cambridge, MA, USA (2018)

[39] Zou, L., Xia, L., Ding, Z., Song, J., Liu, W., Yin, D.: Reinforcement learning to optimize long-term user engagement in recommender systems. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2810–2818 (2019)

[40] Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. arXiv preprint arXiv:1511.06732 (2015)

[41] He, R., McAuley, J.: Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Proceedings of the 25th International Conference on World Wide Web, pp. 507–517 (2016)

[42] Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)

[43] Xian, Y., Fu, Z., Muthukrishnan, S., De Melo, G., Zhang, Y.: Reinforcement knowledge graph reasoning for explainable recommendation. In Proceedings of the

42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 285–294 (2019)

[44] Zhang, Y., Ai, Q., Chen, X., Croft, W.B.: Joint representation learning for top-n recommendation with heterogeneous information sources. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1449–1458 (2017)

[45] Zhou, S., Dai, X., Chen, H., Zhang, W., Ren, K., Tang, R., He, X., Yu, Y.: Interactive recommender system via knowledge graph-enhanced reinforcement learning. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 179–188 (2020)

[46] Hou, Y., Hu, B., Zhang, Z., Zhao, W.X.: Core: simple and effective session-based recommendation within consistent representation space. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1796–1801 (2022)

[47] Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., Guo, M.: 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 (2018)

[48] Wang, X., He, X., Cao, Y., Liu, M., Chua, T.-S.: Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 950–958 (2019)

[49] Zou, D., Wei, W., Mao, X.-L., Wang, Z., Qiu, M., Zhu, F., Cao, X.: 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 (2022)

[50] Wang, P., Fan, Y., Xia, L., Zhao, W.X., Niu, S., Huang, J.: Kerl: A knowledge-guided reinforcement learning model for sequential recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 209–218 (2020)

[51] Han, X., Cao, S., Lv, X., Lin, Y., Liu, Z., Sun, M., Li, J.: Openke: An open toolkit for knowledge embedding. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 139–144 (2018)