User-Centric Path Reasoning towards Explainable Recommendation

Chang-You Tai
Academia Sinica, Taiwan, Taipei
johnnyjana730@gmail.com

Liang-Ying Huang
Academia Sinica, Taiwan, Taipei
lyhuang126@gmail.com

Chien-Kun Huang
Academia Sinica, Taiwan, Taipei
kunkunhuang24@gmail.com

Lun-Wei Ku
Academia Sinica, Taiwan, Taipei
lwku@iis.sinica.edu.tw

ABSTRACT

There has been significant progress in the utilization of heterogeneous knowledge graphs (KG) as auxiliary information in recommendation systems. Reasoning over KG paths sheds light on the user's decision making process. Previous methods focus on formulating this process as a multi-hop reasoning problem. However, without some form of guidance in the reasoning process, such a huge search space results in poor accuracy and little explanation diversity. In this paper, we propose UCPR, a user-centric path reasoning network that constantly guides the search from the aspect of user demand and enables explainable recommendation. In this network, a multi-view structure leverages not only local sequence reasoning information but also a panoramic view of the user's demand portfolio while inferring subsequent user decision-making steps. Experiments on five real-world benchmarks show UCPR is significantly more accurate than state-of-the-art methods. Besides, we show that the proposed model successfully identifies users' concerns and increases reasoning diversity to enhance explainability.

CCS CONCEPTS

• **Information systems → Recommender systems**; • **Computing methodologies → Machine learning**.

KEYWORDS

Recommendation System, Reinforcement Learning, Knowledge Graphs, Explainable Recommendation, Path Reasoning

ACM Reference Format:
Chang-You Tai, Liang-Ying Huang, Chien-Kun Huang, and Lun-Wei Ku. 2021. User-Centric Path Reasoning towards Explainable Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3404835.3462847

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462847

1 INTRODUCTION

Recently, researchers have begun to explore the potential of auxiliary information to improve recommendation performance. Knowledge graphs,[1] hereafter denoted as KG, are one of the best choices for this purpose. Achievement attributed to KG can be observed from many aspects. First, recommendation *accuracy* is improved by item association in KG, which enables the model to exploit underlying user-product relations. Second, multi-hop relations between users and items enhances *explainability*. For example, the recommendation for the movie "Forrest Gump" to user "Alice" is explained by this connection: $\textit{Alice} \xrightarrow{\textit{watched}} \textit{Cast Away} \xrightarrow{\textit{starred}} \textit{Tom Hanks} \xrightarrow{\textit{star}} \textit{Forrest Gump}$, showing that the same actor appears in both movies. The explainability afforded by KG reasoning can prevent recommendation models from relying on post-hoc fallacious explanations [34, 35] and hence increase user trust and satisfaction [25].

Though knowledge graph reasoning has potential, challenges remain. First, effective guidance is lacking in the reasoning process. Sun et al. [23] and Wang et al. [32] adapt recurrent neural networks (RNNs) to calculate scores of qualified candidate paths from the user to the target item. However, the nature of breadth-first exhaustive search makes it difficult to find all potential paths in large-scale KG. Xian et al. [34] propose PGPR, which further utilizes REINFORCE for explicit reasoning in the decision-making process over KG. However, sparse reward signals [35, 37, 43] still lead to poor convergence and thus degraded accuracy.

Second, works [1, 31, 34, 43] on KG reasoning considers none or only part of the diversity within user demand in the historical interaction. Xian et al. [35] address this with CAFE, which models diverse user demand from historical activity data. However, their method fixes user information to predefined user patterns and fails to utilize dynamic reasoning representations [8, 17, 46] to exclude fulfilled demand during reasoning. Without a proper method to model diverse user demand information, the reasoning model generates only general explanations and thus fails to explain this diversity. This also affects explainability since the model fails to consider individual user demand [34].

To model a more panoramic view and reflect dynamic user demand, we consider the fact that user demand is typically not directed toward a single item but is instead represented by groups of entities in the KG for accomplishing a task such as making a cake, going to a party, fishing, etc. Such a consideration thus affects

[1] A knowledge graph is typically described as consisting of entity-relation-entity triplets, where the entity can be an item or an attribute.

the explainability of the recommendation system. In this case, an aggregation of entities fulfills the whole demand; after demand has been partially fulfilled in the reasoning process by any entity in the path, the demand is updated accordingly. We hereafter refer to such demand as a *user demand portfolio*, which constitutes practical user-centric guidance by stressing user demand during the reasoning process and is more likely to reveal items of potential interest for a given user.

Text generation research shows that a multi-view guidance structure helps to model small components under a big umbrella; in our case, generating a user-centric view from the user demand portfolio. Gu et al. [11] indicate that it is crucial to continue to consider the source view during decoding; thus they propose a copying mechanism to model the source view and dynamically fetch relevant pieces of information according to the output sequence view to achieve multi-view guidance. In addition, Serban et al. [20] point out that sequence-to-sequence models focus on local information with the only source of variation being the conditional output distribution. Thus they introduce a hierarchical structure to incorporate a hierarchical view and enhance long-term reasoning. Similar cases are found in our recommendation data. Take for instance user-centric view information: a user has watched *I, Robot*, *Men in Black*, and *The Pursuit of Happyness*, all starring *Will Smith*. We can conclude that *Will Smith* might play an essential role in the user's decision making path. Another example: a user mentions *shampoo* and *hair conditioner* and purchases *Scalp Care* products, suggesting a user demand portfolio focused on hair protection. To collect these user demand portfolios and achieve a user-centric view in the reasoning process, we propose a GNN-based structure in the KG, extending from each user to cover relevant entities.

In this work, we argue the necessity of considering such a user-centric view to provide adequate guidance and diverse user demand information during reasoning. We therefore propose UCPR, a user-centric path reasoning network in which the multi-view structure focuses on both local sequence dependency and user demand information in the path reasoning process; we here term these the *local view* and the *user-centric view*. For each reasoning step, we further propose multi-step refocusing to re-weight the user demand portfolio to identify the most relevant entities for reasoning, after which we propose dynamic representation to update the entities in the portfolio to exclude the fulfilled demand.

We summarize our contributions in this paper:

- We propose a novel user-centric path reasoning network in recommendation to facilitate KG reasoning.
- We propose a dynamic user demand portfolio, showing that user demand is a high-level concept involving a group of entities, which changes with the fulfillment of composite demand, provides effective guidance, and improves the explainability of diversity.
- We enhance the compatibility of two-level networks by multi-step refocusing and dynamic representation updates. To the best of our knowledge, we are the first to incorporate this concept in KG reasoning recommendation.
- Experiments on five real-world datasets with KG of different sizes demonstrate the robustness and superiority of UCPR,[2] which achieves state-of-the-art performance in knowledge path reasoning for recommendation.

[2]We release the codes and datasets at https://github.com/johnnyjana730/UCPR/

2 RELATED WORK

2.1 Multi-view models

Various studies have been conducted on multi-view models, especially in text generation. For example, See et al. [19] propose a hybrid pointer-generator network, which generates words according to the current sequence view while retaining the ability to dynamically copy words from the source view. Bai et al. [3] propose a hierarchical-view architecture to tune the initial order plan at both the word and the attribute levels for textual description generation, which provides the model with more variation to improve the model's expression diversity [20, 21, 36]. Shen et al. [22] propose ml-VAE with a hierarchical view structure that accounts for both higher-level abstract features and lower-level fine-granularity details to generate long, coherent text. These models all adopt multi-view structures to consider global and local—equivalently, coarse and fine—information to produce the final result. In this paper, the user demand portfolio works with the reasoning path. Instead of mixing two types of information together, we check and modify the user portfolio at each reasoning step to ensure it reflects current user demand. To our knowledge, we are the first to propose this multi-view user-centric structure in the KG path reasoning process for recommendation.

2.2 User demand recommendation

Many works attempt to leverage user demand to improve recommendation performance. Among these, some attempt to build accurate user demand representations. For example, An et al. [2] propose LSTUR to jointly consider the user's long- and short-term interests. Hu et al. [14] construct a user-news-topic knowledge graph and propose graph neural networks to extract the user's long-term interests, achieving state-of-the-art performance on news recommendation. To capture user dynamic preferences, Chen et al. [7] propose a time-aware gated recurrent unit to model user dynamic preferences and profile an item by user's review information. The proposed module can provide adaptive recommendation explanations tailored for the users' current preferences. Liu et al. [16] argue that previous memory models may be unable to model user demand over long sessions; they thus improve recommendation via a novel short-term attention/memory priority model that captures users' general interests from long-term sessions and current interests from last-click information. Nevertheless, the goals of all these methods are to capture long-term user demand which is then used as an input feature for precise recommendation. However, they do not consider user demand to guide their reasoning process, that is, a user demand portfolio that is updated when partial demands are fulfilled.

2.3 Leveraging the KG for recommendation

KG-based recommendation models can be divided into two categories. The first is embedding-based methods [4, 5, 29, 33, 39] which combine the entities and relations of a KG into continuous vector spaces and then assist the recommendation system by enhancing the entity representation. However, given these methods' use of only structural information (triplets), their embedding algorithms do not directly utilize the KG, which limits their reasoning ability.

The second category includes path-based models [6, 32, 38, 42], which reason on the underlying connections among users and items in the KG to extract useful information for recommendations. For example, KGAT [31] applies a graph attention network to propagate over the set of entity links to enrich entity representation. However, these methods rely only on post-hoc explanations, as their explanations are not produced according to the reasoning process [34]. Some address this shortcoming using pre-defined connectivity patterns such as metapaths to enable knowledge graph reasoning. The adversarial actor-critic [43], for instance, guides path reasoning with a metapath based framework. However, as defining effective metapaths still necessitates domain knowledge and manual effort, it does not discover unseen or personalized connectivity patterns.

3 PROBLEM FORMULATION

In the KG reasoning problem, the model input consists of the user set $\mathcal{U} = \{u_1, u_2 ...\}$, the item set $\mathcal{V} = \{v_1, v_2 ...\}$, and the observed interaction set $\mathcal{V}_u = \{v | y_{uv} = 1\}$, which contains all observed interactions between user u and item v in the training set. Also, the knowledge graph $\mathcal{G}$ is comprised of entity-relation-entity triplets $\{(e^h, r, e^t) | e^h, e^t \in \mathcal{E}, r \in \mathcal{R}\}$. Triplet (e^h, r, e^t) describes relations r from head entity e^h to tail entity e^t, and $\mathcal{E}$ and $\mathcal{R}$ denote the set of entities and relations in $\mathcal{G}$. In addition, we follow previous work [31, 34] in incorporating user-item information into the KG. Finally, given a user u, our model outputs the recommended item set $\widehat{\mathcal{V}_u} \subseteq \mathcal{V}$ and a reasoning path for each recommended item $\tau_{u,\widehat{v}_u}$, where $\widehat{v}_u \in \widehat{\mathcal{V}_u}$ is a t-hop path in $\mathcal{G}$ which connects user u with $\widehat{v}_u$: $\tau_{u,\widehat{v}_u} = [u \xrightarrow{r_1} e_1 \xrightarrow{r_2} .. \xrightarrow{r_{t-1}} e_{t-1} \xrightarrow{r_t} \widehat{v}_u]$, $t = 1, 2, ..., \mathrm{T}$.

4 UCPR

4.1 MDP Environment

Formally, the MDP environment is defined by a tuple $(\mathcal{S},\mathcal{A},\delta,\rho)$ [9], where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, δ is the state transition function, and ρ is the reward function.

States. State $s_t \in \mathcal{S}$ is the search status at step t and is defined as a tuple $(u, e_t, e_{t-1}, \mathcal{V}_u)$, where u denotes the user, e_t is the entity reached by the agent at step t, e_{t-1} is the previously visited entity, and the set of multiple answers $\mathcal{V}_u = \{e_{a_1}, ..., e_{a_n}\}$ is a set of interacted items for user u. Note that at step $t = 0$, user u is set as the starting entity, where $s_0 = (u, u, \varnothing, \mathcal{V}_u)$.

Actions. The set of candidate actions $\mathcal{A}(s_t)$ at time step t is based on state s_t. The action space $\mathcal{A}(s_t) = \{(r', e') \mid (e_t, r', e') \in \mathcal{G})\}$ denotes all possible outgoing edges of current entity e_t, where e' is the next entity in the path and r' is the relation that connects e_t with e'. At each time step t, the agent outputs an action $a_t = (r_{t+1}, e_{t+1}) \in \mathcal{A}(s_t)$. Note that we follow Xian et al. [34] in adding reverse edges and action pruning.

Transitions. The transition function $\delta : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ determines the transitions to the next state s_{t+1} based on state s_t and action a_t: $\delta(s_t, a_t) = s_{t+1} = (u, e_{t+1}, e_t, \mathcal{V}_u)$.

Rewards. After the terminal transition, if location e_T matches any of the answers $\mathcal{V}_u$, a positive reward of +1 is given, or 0 otherwise: $\rho(s_\mathrm{T}) = \mathbb{1}\{e_\mathrm{T} \in \mathcal{V}_u\}$.

4.2 Multi-View Policy Network

The multi-view policy network contains local- and user-view reasoning networks. Figure 1 and Algorithm 1 illustrate how the agent constructs a decision trajectory. The proposed policy network learns a path-finding policy π that maximizes the expected cumulative reward during the reasoning process. In particular, it calculates the probability distribution of action a_t based on state s_t and its action space $\mathcal{A}(s_t)$: $p(a_t|s_t, \mathcal{A}(s_t)) = \pi_\theta(a_t, s_t, \mathcal{A}(s_t))$; the policy network π_θ is defined as

$$\pi_\theta(a_t, s_t, \mathcal{A}(s_t)) = \sigma(\mathbf{A}^l_{s_t} \times \mathbf{h}^l_t + \lambda(\mathbf{A}^u_{s_t} \times \mathbf{h}^u_t)), \tag{1}$$

where $\mathbf{h}^l_t$ and $\mathbf{h}^u_t$ are the user reasoning information at the local- and user-centric levels, respectively, as discussed below. The trainable parameter λ integrates the local and user-centric view information, and σ is the softmax operator. $\mathbf{A}^l_{s_t} \in \mathbb{R}^{|\mathcal{A}(s_t)| \times 2d}$ and $\mathbf{A}^u_{s_t} \in \mathbb{R}^{|\mathcal{A}(s_t)| \times d}$ denote the local and user-centric action set matrix, respectively. At time step t, we obtain action embedding $\mathbf{a}^l_t = (\mathbf{r}_t, \mathbf{e}_t)$ and $\mathbf{a}^u_t = (\mathbf{e}_t)$ from action $a_t = (r_t, e_t)$ with a lookup layer, respectively, after which we stack all possible action embeddings to obtain $\mathbf{A}^l_{s_t}$ and $\mathbf{A}^u_{s_t}$.

Algorithm 1: Multi-view Path Reasoning

```
Input: Interaction set V_u, knowledge graph G(E, R), T;
Output: Trajectory τ
HVPR:
    s_0 ← (u, u, ∅, V_u)
    e_0 ← u
    for t = 0, ..., T − 1 do
        h^l_t ← LV(h, a)
        h^u_t ← UV(e_t)
        Calculate probability distribution of action space
        p(a_t | s_t, A(s_t)) = σ(A^l_{s_t} × h^l_t + λ(A^u_{s_t} × h^u_t))
        Sample action a_t = (r_{t+1}, e_{t+1})
        e ← f^{DR}(e_{t+1}; e), ∀e ∈ P_u
        s_{t+1} ← (u, e_{t+1}, e_t, V_u)
    return τ = s_0, a_0, s_1, a_1, ..., s_T
```

4.3 Local-View Reasoning Network

The local-view reasoning network encodes sequential information using LSTM [13]. The search history consists of a sequence of actions taken by the agent. An action chosen by the agent at time step t is $a_t \in \mathcal{A}(s_t)$ and its action embedding $\mathbf{a}^l_t = (\mathbf{r}_t, \mathbf{e}_t)$ are encoded by the LSTM network as

$$\mathbf{h}^l_t = \mathbf{LV}(\mathbf{h}, \mathbf{a}) = \begin{cases} \mathrm{LSTM}(0, [\mathbf{r}_0; \mathbf{u}]), t = 0 \\ \mathrm{LSTM}(\mathbf{h}^l_{t-1}, \mathbf{a}^l_{t-1}), t > 0, \end{cases} \tag{2}$$

where $\mathbf{a}^l_t \in \mathbb{R}^{2d}$, $[;]$ denotes vector concatenation, and $\mathbf{r}_0$ and $\mathbf{u}$ denote the embedding of the initial relation and user u.

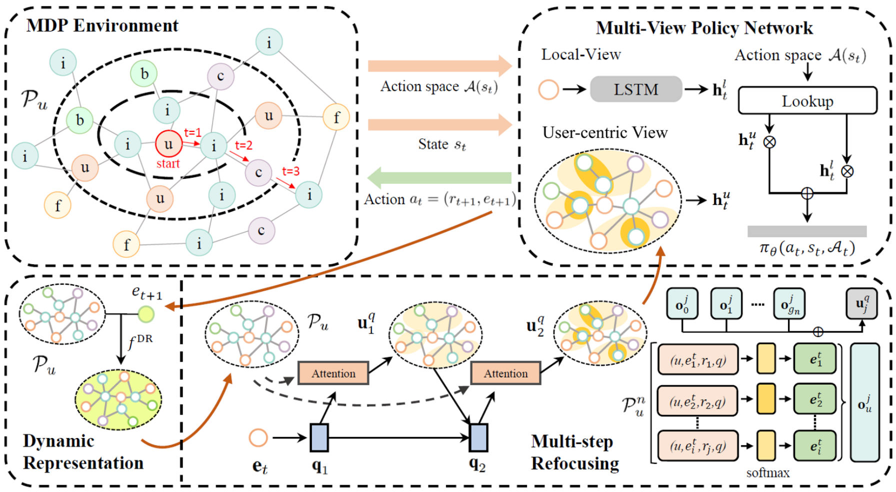

Figure 1: Framework of proposed user-centric path reasoning network

4.4 User-centric view Reasoning Network

The user-centric view reasoning network contains two components: a multi-step refocusing module and a dynamic representation module. The former generates an exact user demand portfolio to represent user-centric level information, as described in Algorithm 2, and the latter excludes fulfilled demand by updating the whole portfolio after making a step.

4.5 Multi-step Refocusing Module

To precisely highlight the relevant entities in the portfolio set, we propose multi-step refocusing, in which the query entity recursively retrieves entities in the portfolio set to generate the final user portfolio information. At path-finding step t, the entity vector $\mathbf{e}_t$ chosen by the agent acts as the initial query $\mathbf{q}_1$. During the refocusing step j, the propagated user portfolio $\mathbf{u}_j^q$ is generated according to the current query vector $\mathbf{q}_j$ and is then updated to generate the next query vector $\mathbf{q}_{j+1}$. A simple one-step refocusing pathway is illustrated as $\mathbf{q}_j \rightarrow \mathbf{u}_j^q \rightarrow \mathbf{q}_{j+1}$, which is performed g_r times to generate the final user portfolio information $\mathbf{h}_t^u = \mathbf{u}_{g_r}^q$. This is described below.

Algorithm 2: User-centric View Reasoning

```
UV (e_t):
    q_1 ← e_t
    for j = 1, ..., g_r do
        for n = 1, ..., g_n do
            o_n^j = Σ_{(e_i^h, r_i, e_i^t) ∈ P_u^n} k_i^j e_i^t
        u_j^q = W_o [o_0^j; o_1^j; ...; o_{g_n}^j] + b_o
        q_{j+1} = W_n [u_j^q; q_j]
    return u_{g_r}^q
```

Portfolio set To initialize, the set of items $\mathcal{V}_u = \{v | y_{uv} = 1\}$ that user u has interacted with is treated as the starting point in $\mathcal{G}$, which is then explored along the relations to construct the portfolio set $\mathcal{P}_u$ as

$$\mathcal{E}_u^n = \{e^t | (e^h, r, e^t) \in \mathcal{G}, e^h \in \mathcal{E}_u^{n-1}\}, \tag{3}$$

where $\mathcal{E}_u^0 = \mathcal{V}_u$, and $\mathcal{E}_u^n$ records the n-hop entities linked from entities at the previous $(n-1)$-th hop, and

$$\mathcal{P}_u^n = \{(e^h, r, e^t) | (e^h, r, e^t) \in \mathcal{G}, e^h \in \mathcal{E}_u^{n-1}\}, \tag{4}$$

where $\mathcal{P}_u^n$ is the portfolio set at hop n, $n = 1, 2, ..., g_n$. Note that $\mathcal{E}_u^n$ contains only tail entities and $\mathcal{P}_u^n$ is the set of knowledge triplets. The number of portfolio hops is g_n. In a real-world knowledge graph, the size of $N(e)$ varies significantly. To control computational efficiency, we adopt a fixed-size strategy [28]: at each hop n, we sample only a fixed number of relevant entities in portfolio set $\mathcal{P}_u^n$, where $|\mathcal{P}_u^n| = p_m$ and p_m is the portfolio set sampling size.[3]

4.5.1 Portfolio Propagation. At refocus step j, to generate the propagated user portfolio $\mathbf{u}_j^q$, we first define the corresponding relevance probabilities k_i^j, which are used to weight each tail entity e_i^t in the portfolio set $\mathcal{P}_u^n$ at hop n. When calculating relevance probabilities k_i^j, we assume that the user, relation, and previously chosen entity are needed to retrieve the relevant entities in the portfolio set:

$$k_i^j = \sigma(\mathbf{W}_k^j \times \tanh(\mathbf{W}_u^j \mathbf{u} + \mathbf{W}_g^j \mathbf{R}_i \mathbf{e}_i^t + \mathbf{W}_s^j \mathbf{R}_t \mathbf{e}_i^t + \mathbf{W}_q^j \mathbf{q}_j])), \tag{5}$$

where σ is the softmax operator. $\mathbf{q}_j \in \mathbb{R}^d$ denotes the query vector. $\mathbf{u} \in \mathbb{R}^d$, $\mathbf{e}_i^t \in \mathbb{R}^d$, $\mathbf{R}_i \in \mathbb{R}^{d \times d}$, and $\mathbf{R}_t \in \mathbb{R}^{d \times d}$ are the embeddings of user u, tail e_i^t, relation r_i, where $(e_i^h, r_i, e_i^t) \in \mathcal{P}_u^n$ at hop n, and current step relation r_{t-1}. The relation space embedding $\mathbf{R}_i$ and $\mathbf{R}_t$ are used to calculate the relevance of entity representation $\mathbf{e}_i^t$ and the query vector; $\mathbf{W}_k^j$, $\mathbf{W}_u^j$, $\mathbf{W}_g^j$, $\mathbf{W}_s^j$, and $\mathbf{W}_q^j$ are trainable parameters.

Given the corresponding relevance probabilities k_i^j, we sum the user demand in the portfolio set $\mathcal{P}_u^n$ at hop n to generate the user response $\mathbf{o}_n^j$ at hop n as:

$$\mathbf{o}_n^j = \sum_{(e_i^h, r_i, e_i^t) \in \mathcal{P}_u^n} k_i^j \mathbf{e}_i^t, n = 1, 2, ..., g_n. \tag{6}$$

To integrate user responses $\mathbf{o}_n^j$, we use MLP to generate the propagated user portfolio $\mathbf{u}_j^q \in \mathbb{R}^d$ at the j-th step as:

[3] We discuss the performance changes when p_m is varied.

$$\mathbf{u}_j^q = \mathbf{W}_o[\mathbf{o}_0^j; \mathbf{o}_1^j; \ldots, \mathbf{o}_{g_n}^j] + \mathbf{b}_o, \quad (7)$$

where $\mathbf{W}_o$ and $\mathbf{b}_o$ are trainable parameters.

4.5.2 Update query vector. To update the next query vector, we apply a linear transformation function using both the propagated user portfolio $\mathbf{u}_j^q$ and the query vector $\mathbf{q}_j$:

$$\mathbf{q}_{j+1} = \mathbf{W}_n[\mathbf{u}_j^q; \mathbf{q}_j], \quad (8)$$

where $\mathbf{W}_n$ are trainable parameters. After the initial query vector $\mathbf{q}_1$ is set to $\mathbf{e}_t$, this process forms a cycle completing one refocus step, which is performed g_r times to generate the final user portfolio information $\mathbf{h}_t^u = \mathbf{u}_{g_\mathrm{r}}^q$.

4.6 Dynamic Representation Module

To update all of the entities in the portfolio to exclude the fulfilled demand during the reasoning process, we propose dynamic representation to change the perspective of entities in the portfolio set. For this purpose, we define a function f^{DR}. Given the agent-selected entity vector $\mathbf{e}_t$ and the entity representation $\mathbf{e}$, $e \in \mathcal{P}_u$, f^{DR} generates the new entity representation $\tilde{\mathbf{e}}$ as

$$\tilde{\mathbf{e}} = f^{\mathrm{DR}}(\mathbf{e}_t; \mathbf{e}) = \mathbf{W}_f[\mathbf{e}_t; \mathbf{e}] + \mathbf{b}_r, \quad (9)$$

where $\mathbf{W}_f$ and $\mathbf{b}_r$ are trainable parameters. Here we keep f^{DR} in its general form; it can be replaced by any state-of-the-art method.

4.7 Optimization

To optimize UCPR, the policy network π_θ is trained by maximizing the expected reward over all queries for user u:

$$J(\theta) = \mathbb{E}_{(u,\mathcal{V}_u)\in \mathrm{D}}[\mathbb{E}_{a_1,\ldots,a_\mathrm{T}\sim\pi_\theta}[\sum_{t=1}^{\mathrm{T}} \gamma^{t-1}\rho(s_\mathrm{T}|u)]], \quad (10)$$

where γ, a discount factor, is set to 1, and D is the training dataset.

4.8 Time complexity

Per batch, the time cost for UCPR mainly comes from executing the local-view and user-centric view reasoning networks. The local-view network incurs a time complexity of LSTM $O(C^L)$, and the user-centric view reasoning network incurs a computational complexity of $O(g_r(g_n * p_m * d^2))$ to calculate the relevance probability k_i^j for total of p_m entities, g_n layers, and g_r reasoning time. Thus, the overall training complexity of UCPR is $O(\mathrm{T}(C^L + g_r(g_n * p_m * d^2) + |\mathcal{A}(s_t)|d))$, where T is reasoning path length and $|\mathcal{A}(s_t)|$ is the number of possible actions.

We also compare the time cost of UCPR with path reasoning models such as PGPR and CAFE. For PGPR, the policy network's time cost is $O((sd)^2 + \mathcal{A}(s_t))$, where s is the history steps. Thus, the total time cost is $O(\mathrm{T}((sd)^2 + |\mathcal{A}(s_t)|))$; CAFE incurs a time complexity of $O(MTd^2)$, where M is the number of user-centric patterns; we conducted experiments to compare the training speed of UCPR and other methods on an GTX-1080 GPU. Empirically, UCPR, PGPR, and CAFE use 223.4s, 100.3s, and 90.7s, respectively, to iterate all training user-item pairs in the Cell Phones dataset.

5 EXPERIMENTS AND RESULTS

5.1 Datasets

We utilized five real-world datasets: MovieLens-1M (ML-1M), Amazon Book (Book), Beauty, Clothing, and Cell Phones [12, 28, 30, 31]; For ML-1M, its KG were built using Microsoft Satori where the confidence level was set to greater than 0.9. The Book KG was built using title matching, as described in Zhao et al. [44]. Other datasets followed previous studies [1, 41, 44]. In addition, all datasets contained review text information[4] except for ML-1M. To ensure dataset quality, we applied an α-core setting [31], that is, we retained users and items with at least α interactions. For Book, α was set to 20. For Beauty, Clothing, and Cell Phones, due to their relatively low average number of interactions, α was set to a smaller value of 8. We randomly sampled 60% of each user's interactions as the training set and took the remaining 20% and 20% as the evaluation and test sets. Table 1 shows the dataset statistics.

5.2 Baseline Models

We compared UCPR with the following baselines: **BPR** [18], a Bayesian personalized ranking model that learns latent embeddings of users and items; **DeepCoNN** [45], a text-based convolutional recommendation model which learns user and item representations jointly based on reviews; and **JRL** [41], a joint representation learning model that makes use of multiple sources of information, for instance text and ratings, to train a deep neural network. In the ML-1M dataset, we used only rating information as input, whereas the other datasets were trained on both text and ratings. Other baselines were **CKE** [40], a representative embedding-based method in which only structural knowledge information is used; **KGAT** [31], a path-based method which utilizes the GAT mechanism to discriminate neighbor importance; **MVIN** [24], a path-based method that focuses on user-item and entity-entity views to enhance item representation; **HeteroEmbed** [1], a post-hoc path reasoning approach that adopts TransE embeddings for recommendation; **PGPR** [34], a KG path reasoning model which adopts a policy-based reinforcement learning method; and **CAFE** [35], a state-of-the-art KG path reasoning model that generates user profiles as coarse sketches of user behaviors, followed by a path-finding process to derive reasoning paths for recommendations. To generate a user-centric pattern set, we adopted a random walk based method per Xian et al. [35].[5] Note that we did not include ADAC [43] as a baseline because we lack domain experts to generate well-defined metapaths.

5.3 Experimental Setup

We evaluated the proposed model's recommendation accuracy and explainability. To better understand model performance, we conducted an ablation study to investigate the effectiveness of model components. We use Recall, NDCG, and Hit Ratio (HR) to evaluate the recommendation performance of UCPR. We evaluate the metrics based on the top-10 recommended items for each user in the test set, and report the results in percentages.

To evaluate the reasoning paths' explainability, we first follow Fu et al. [10] in conducting a review matching experiment in which

[4] http://jmcauley.ucsd.edu/data/amazon

[5] We reproduced the CAFE code since their code is not available.

Table 1: Dataset statistics

	Clothing	Cell Phones	Beauty	ML-1M	Book
Users / Items	10,228 / 22,298	6,548 / 10,029	8,300 / 12,007	6,036 /2 ,445	6,969 / 9,854
Interactions	114,302	75,070	118,606	753,772	522,706
Avg user clicks / clicked items	8.25 / 3.97	8.46 / 5.79	10.44 / 7.38	124.9 /308.3	79.3 / 56.1
Avg 3-hop products (search space) / ans. covering rate	2469.5 / 58.5%	1540.0 / 71.6%	1446.7 / 67.5%	1092.8 / 95.8%	4136.3 / 90.8%
Entities / Relation types	425,528 / 8	163,249 / 8	224,074 / 8	182,011 / 12	113.487 / 39
Triples	10,671,090	6,299,494	7,832,720	1,241,995	2,557,746

Table 2: Overall recommendation effectiveness of proposed method compared to other baselines on five datasets. (Note that DeepCoNN is not applicable on ML-1M as it contains no review text information.)

Dataset	Clothing			Cell Phones			Beauty			ML-1M			Book		
Metrics	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG	HR
BPR	0.883	0.536	2.483	3.697	2.248	9.423	3.230	2.191	10.084	6.522	8.144	46.154	2.803	2.790	18.012
DeepCoNN	1.613	1.123	3.097	4.017	2.729	10.571	3.442	2.847	11.021	-	-	-	3.233	3.737	20.737
JRL	1.787	1.234	4.517	4.328	3.220	11.042	4.203	3.675	13.214	8.306	9.920	50.701	3.430	3.538	22.092
CKE	1.645	0.959	4.742	4.891	3.152	13.821	4.075	2.911	12.723	11.010	11.879	57.795	4.360	4.338	26.238
KGAT	1.784	1.225	4.840	5.332	3.660	14.111	4.811	3.418	15.048	11.017	12.251	58.650	4.905	4.791	27.228
MVIN	1.709	0.933	3.989	5.527	3.847	12.462	4.749	3.722	14.386	13.715	17.167	65.573	5.027	4.980	28.236
HeteroEmbed	1.801	1.236	5.121	5.681	4.085	14.512	5.015	3.954	15.621	9.051	10.539	52.367	4.034	3.985	23.137
PGPR	3.758	2.910	10.110	7.360	5.711	18.677	6.657	5.577	18.590	10.449	11.557	52.214	4.993	4.884	28.671
CAFE	3.813	3.054	10.685	8.011	6.051	19.326	5.578	4.523	16.842	11.539	12.163	55.661	4.836	4.627	27.714
UCPR	**4.581***	**3.226***	**11.576***	**8.606***	**6.380***	**21.358***	**8.812***	**6.999***	**24.289***	**14.306***	**18.637***	**67.385***	**8.639***	**9.115***	**41.900***

Note: * indicates statistically significant improvements over the best baseline by an unpaired two-sample t-test with p-value = 0.01.

we match the entities in the reasoning paths with the ground-truth words mentioned in the ground-truth review with the metrics Recall and NDCG based on the top-10 matched entities, i.e., to reason about the aspects valued by users. Second, since explanation diversity affects the explainability of the reasoning paths [34], to evaluate the diversity of path distribution, we follow Zhao et al. [43] in using Simpson's index of diversity (SID) as a metric to quantify the difference in species distributions. Compared to Shannon entropy, SID is less sensitive to species size because it accounts for both richness and evenness. We evaluate both word and path diversity to calculate the explanation diversity of a model. For path diversity, we randomly select two individual user-item paths and use SID to measure the probability that these two paths belong to the same user-item pattern. Word diversity is evaluated similarly. Formally, SID is defined as

$$\mathrm{SID}(R, N) = 1 - \frac{\sum_{i=1}^{R} n_i(n_i - 1)}{N(N-1)}. \tag{11}$$

Take for example path SID, where R represents the number of path patterns generated by the model, n_i denotes the number of such paths belonging to the i-th path pattern, and N is the total number of user-item paths: the SID value ranges from 0 to 1, where higher values represent higher diversity.

For UCPR, for Clothing, $g_{\mathrm{n}} = 2$, $g_{\mathrm{r}} = 2$, $p_{\mathrm{m}} = 64$, $\lambda = 0.5$; for Cell Phones, $g_{\mathrm{n}} = 3$, $g_{\mathrm{r}} = 3$, $p_{\mathrm{m}} = 64$, $\lambda = 0.5$; for Beauty, $g_{\mathrm{n}} = 3$, $g_{\mathrm{r}} = 2$, $p_{\mathrm{m}} = 32$, $\lambda = 0.7$; for ML-1M, $g_{\mathrm{n}} = 2$, $g_{\mathrm{r}} = 4$, $p_{\mathrm{m}} = 64$, $\lambda = 2.5$; and for Book, $g_{\mathrm{n}} = 2$, $g_{\mathrm{r}} = 3$, $p_{\mathrm{m}} = 32$, $\lambda = 0.3$. For UCPR and the baselines, the embedding size d was set to 16 for Clothing, Cell Phones, and Beauty, and 32 for ML-1M and Book with state vectors of 32 and 64 respectively. Note that the state vectors were 64 and 128 for PGPR, as PGPR adopts 1-step histories. For CAFE, the weighting factor for the ranking loss is set to 8. For the MDP environment, we mainly followed Xian et al. [34]. We set the maximum length T = 3 for reasoning path τ. For testing, probabilistic beam search was used to reason over paths; we set the sampling sizes to $K_1 = 10$, $K_2 = 15$, $K_3 = 1$ for Clothing, Cell Phones, and Beauty, and $K_1 = 8$, $K_2 = 3$, $K_3 = 4$ for ML-1M and Book.[6]

Table 3: Comparison of explainability

Dataset	Clothing		Cell Phones		Beauty	
Metrics	Recall	NDCG	Recall	NDCG	Recall	NDCG
PGPR	0.427	8.836	0.480	7.496	0.348	6.438
CAFE	0.452	9.034	0.513	7.682	0.318	5.891
$\mathrm{UCPR}_{\mathrm{w/o\,UV}}$	0.595	10.394	0.558	8.117	0.485	7.168
UCPR	**0.754**	**11.292**	**0.854**	**11.356**	**0.685**	**9.272**

Table 4: Comparison of explanation diversity

Dataset	Clothing		Cell Phones		Beauty	
SID	word	path	word	path	word	path
PGPR	0.356	0.203	0.457	0.218	0.211	0.134
CAFE	0.491	0.193	0.536	0.257	0.289	0.203
$\mathrm{UCPR}_{\mathrm{w/o\,UV}}$	0.681	0.287	0.511	0.261	0.391	0.183
UCPR	**0.802**	**0.660**	**0.765**	**0.684**	**0.694**	**0.685**

5.4 Results and Discussion

5.4.1 Recommendation Accuracy. Table 2 shows the results of UCPR and the baselines. We make the following observations.

UCPR outperforms state-of-the-art KG path reasoning baselines CAFE, PGPR, and HeteroEmbed by a large margin. For example, compared to CAFE, it yields NDCG performance gains of 20.14%, 5.43%, 54.74%, 53.23%, and 97.00% on Clothing, Cell Phones, Beauty, ML-1M, and Book, respectively; similar trends are observed for Recall measurement. These results indicate that the proposed user-centric view guidance helps UCPR to better identify user demand. Models such as PGPR suffer from an inefficient search strategy and sparse reward signals. Also, compared to UCPR, CAFE only uses fixed user patterns generated from pre-sampled paths to derive reasoning paths and does not utilize dynamic and fine-grained user demand information—such as that represented by our user demand portfolio—to exclude fulfilled demand during reasoning, thus yielding inferior performance. Also, note that CAFE does not

[6]Parameters optimized per Xian et al. [34].

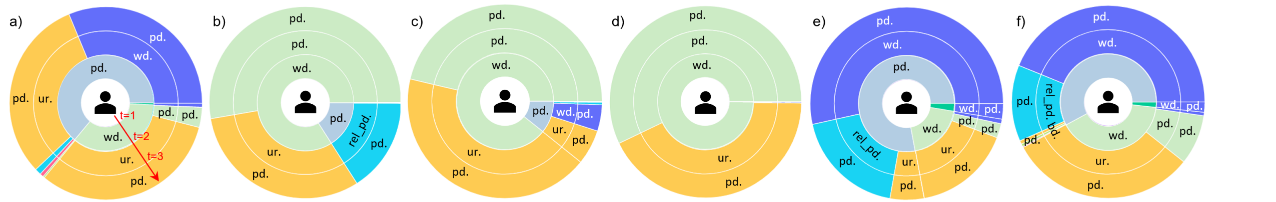

Figure 2: Sunburst plots for explanation diversity. We compare variations between a) UCPR, b) UCPR$_{\mathrm{w/o\,UV}}$, c) CAFE, d) PGPR, e) UCPR (p'_{m}), and f) UCPR (p^{*}_{m}) based on Cell Phone datasets. The number of rings is counted from inside to outside, representing the hops of the path. The *i*-th ring captures the frequency distribution over the reasoning path's patterns at the *i*-th step. "d.", "prod.", "rel pd.", and "ur." denote "word", "product", "related product", and "user".

consistently outperform PGPR, possibly because CAFE relies heavily on the quality of its user-centric pattern set, and because of the noise inherent to such random walk based methods.

UCPR outperforms path-based baselines such as MVIN and KGAT on all datasets, indicating that reasoning paths over the knowledge graph yields substantial benefits for product recommendation. Another reason for the superior performance is the reduced search space. For example, in Table 1, for the Book dataset, 3-hop candidate products amount for half of the total products but retain more than 90% of the answers. Thus, it is more efficient to find answers over reasoning paths in the KG. Note that UCPR outperforms MVIN just barely for ML-1M, but by a huge margin on datasets with high KG node out-degrees such as Cell Phones and Beauty. These results are not surprising. Note that path reasoning methods only recommend items connected to previously purchased items, i.e., along the path; this limited connectivity affects performance.

5.4.2 Explainability and Diversity. We first evaluate the explainability of the proposed model via review matching. Table 3 compares the explainability of path reasoning methods PGPR, UCPR$_{\mathrm{w/o\,UV}}$, removing the user-centric view (UV), and UCPR. We find that UCPR outperforms the baselines. For example, compared to CAFE, it yields NDCG performance gains of 25.0%, 47.8%, and 57.4% on Clothing, CellPhones, and Beauty, respectively. This indicates that the proposed user-centric view improves the explainability of the reasoning paths. Section 5.5 describes a case study on how user-centric view guidance improves explainability.

Next we illustrate the diversity of the proposed reasoning paths to further evaluate the explainability. Table 4 and Fig. 2 show the results. First, we observe that the proposed model yields higher diversity than other baseline models in terms of word and path SID scores, indicating that highlighting diverse user demands during path reasoning via a user-centric view increases the overall explanation diversity of the decision paths. Although CAFE generates path patterns according to each user's prominent path weight, it still fails to generate diversity patterns; this is mainly attributed to the quality and diversity of each user's user-centric pattern set. The sunburst plots in Fig. 2 further illustrate the diversity level of different approaches. Compared to PGPR and UCPR$_{\mathrm{w/o\,UV}}$, UCPR generates more diverse patterns of reasoning paths, e.g. *user* $\longrightarrow$ *product* $\longrightarrow$ *word* $\longrightarrow$ *product* and *user* $\longrightarrow$ *product* $\longrightarrow$ *related product* $\longrightarrow$ *product* instead of being confined to the pattern of *user* $\longrightarrow$ *word* $\longrightarrow$ *product* $\longrightarrow$ *product* as with PGPR and UCPR$_{\mathrm{w/o\,UV}}$.

5.4.3 Convergence. We investigate the effect of the selected entity size on model performance. We denote p_{m} as the size of the portfolio sampling set. Figure 3 shows that when p_{m} is large, e.g., 32 or 16, UCPR not only outperforms UCPR$_{\mathrm{w/o\,UV}}$ but also converges faster than with the Cell Phones and Beauty datasets. We conclude that the user-centric view enriches user demand portfolios at each reasoning step by finding the relevant entities for the reasoning process and thus increasing the model's speed of convergence.

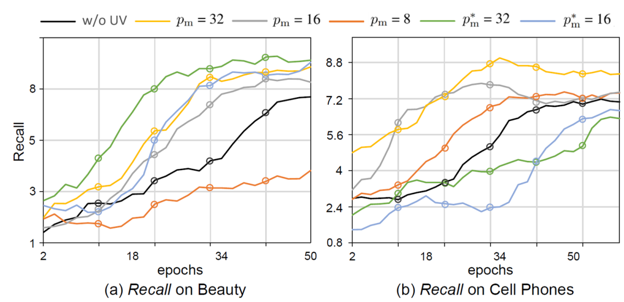

Figure 3: Convergence comparison

Here, p^{*}_{m} and p'_{m} denote the portfolio set without *product*- and *word*-type entities. Fig. 3 shows for Cell Phones, UCPR (p^{*}_{m}), which removes product entities, takes longer time to converge compared to UCPR (p_{m}). Note this is not true for Beauty. From our observations, we find for the Cell Phones dataset, electrical products have a greater impact on other relational items, e.g., their accessories. For example, if users buy an iPhone 5 and it is included in the portfolio set, the model will more easily reason to a phone case designed for the iPhone 5 that they can only buy rather than one for an iPhone 4; likewise for lightning cables for charging. However, the bounding between products in Beauty is not strong: users can buy the eye shadow of one brand and the lipstick of another.

In addition to the impact on convergence, diverse entity types in the portfolio set may also benefit model explainability. Zhao et al. [43] have pointed out different paths have different persuasive levels on explainability. We find that selected entity types in the user portfolio set can help the model generates specific reasoning types of paths. For Cell Phones dataset, figure 2(a) shows that compared to UCPR (p^{*}_{m}) figure 2(f), since UCPR (p_{m}) includes more *product* entities, it tends to generate more path patterns containing *product* entities such as *user* $\longrightarrow$ *product* $\longrightarrow$ *word* $\longrightarrow$ *product*; on the other hand, compared to UCPR (p'_{m}) figure 2(e), in Fig. 2(a), UCPR (p_{m}) generates more path patterns containing *word* entities, e.g., *user* $\longrightarrow$ *word* $\longrightarrow$ *user* $\longrightarrow$ *product*. These examples illustrates that different entity types in the portfolio help UCPR to generate diverse reasoning paths for better persuasiveness.

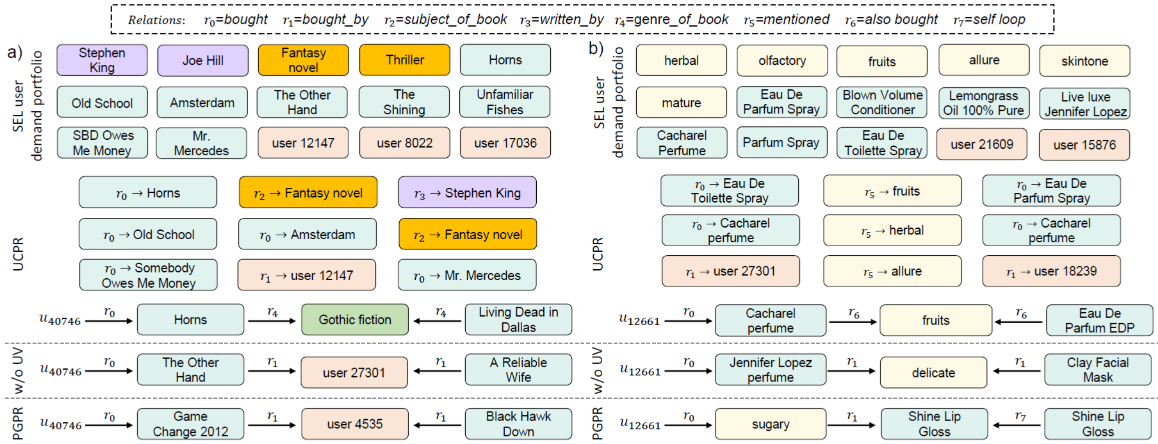

Figure 4: Multi-view reasoning paths. At time step t, we select entities with the highest weight in the user demand portfolio.

Table 5: Recall of UCPR given preference set hops g_n.

g_n hops ($p_m = 32$)	1	2	3
Clothing	4.414	4.430	4.457
Cell Phones	8.337	8.391	8.532
Beauty	8.081	8.557	8.738

Table 6: Performance given refocus steps g_r

Abla.	Clothing		Cell Phones		Beauty	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
$g_r = 1$	4.404	3.132	8.227	6.092	8.296	6.610
$g_r = 2$	4.510	3.183	8.452	6.207	8.587	6.756
$g_r = 3$	4.502	3.176	8.532	6.222	8.729	6.930

Table 7: Recall of UCPR given integration weight λ

λ	0.1	0.3	0.5	0.7	0.9
Clothing	4.410	4.510	4.582	4.493	4.285
Cell Phones	8.231	8.532	8.541	8.384	8.381
Beauty	8.434	8.664	8.689	8.775	8.538

Table 8: Recall of UCPR given embedding size d

d	8	16	32	64
Clothing	3.861	4.557	4.424	4.285
Cell Phones	7.058	8.646	8.576	8.185
Beauty	8.145	8.751	8.686	8.386

5.4.4 Parameter Sensitivity. First, the impact of *number of hops* g_n is shown in Table 5: setting g_n to 3 yields the best performance for these datasets. We attribute this to the fact that the coverage of user demand, i.e., the completeness of the user demand portfolio, increases as g_n grows [28], although larger g_n may reduce computational efficiency. Second, Table 7 represents the performance of UCPR given different values of the *integration weight* λ . Performance is improved when λ is set to 0.5, showing user demand information plays a vital role. However, for larger λ values, performance drops, suggesting that the ratio of local- and user-centric level information is a trade-off. Last, Table 8 shows that increasing *dimension of embedding size* d initially boosts performance, since a larger d can encode more information in the reasoning process; excessive d values, e.g, $d > 16$, however, result in overfitting.

5.5 Ablation Study

Table 9 shows the ablation experiment results.

Local-view and User-centric Reasoning Network. Removing the local-view reasoning network (UCPR$_{\mathrm{w/o\,LV}}$) and removing the user-centric view (UV) reasoning network (UCPR$_{\mathrm{w/o\,UV}}$) both yield worse performance than UCPR for all datasets. This shows that local sequence information and the user demand portfolio both improve recommendation results.

We conducted a case study to understand how the user-centric view guides the reasoning process. Figure 4 shows UCPR generates reasoning paths according to the highlighted entities in the user demand portfolio. For example, based on the selected user demand portfolio in Fig. 4(a) we infer users prefer fantasy, mystery, and thriller novels. At time step $t = 2$, the highlighted entities *Fantasy*, the book's genre, and *Amsterdam*, the title of a mystery novel, help to select the next-step reasoning (*genre of book*, *Gothic fiction*). In Fig. 4(b), we infer that the user cares about perfumes and their fragrances. At time step $t = 2$, the highlighted entities *fruits* and *allure*—words mentioned by the user—guide the reasoning process to select the next action (*describe*, *fruits*), exactly the same words highlighted via the user-centric view. Compared to UCPR, UCPR$_{\mathrm{w/o\,UV}}$ and PGPR select *A Reliable Wife*, a historical novel, *Back Hawk Down*, a historical novel, *Clay Facial Mask*, and *Shine Lip Gloss*, which fail to match user demands. These examples show that the user-centric view highlights relevant entities in the user demand portfolio to directly guide the reasoning process in each step, thus improving recommendation performance and yielding paths that reflect diverse user demand. Also, in Fig. 4(b), the model highlights words mentioned by the user and generates the next action accordingly. It is easier for the model to capture ground-truth words mentioned in the user review, thus increasing explainability.

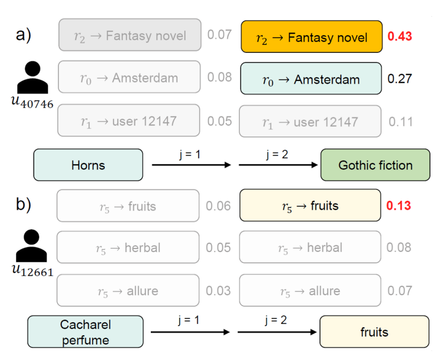

Figure 5: Visualization of learned attention in multi-step refining process, where j denotes the refining step

Table 9: UCPR ablation study results. The multi-view policy network includes local-view (LV) and user-centric view (UV) reasoning networks, and dynamic representation (DR) is used to update all entities in the portfolio.

Ablation	Clothing		Cell Phones		Beauty		ML-1M		AZ-Book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
NA	**4.451***	**3.166***	**8.606***	**6.380***	**8.812***	**6.999***	**14.306***	**18.637***	**8.639***	**9.115***
w/o UV	4.237	3.055	8.171	5.913	8.270	6.568	13.161	17.416	7.332	7.450
w/o LV	2.796	2.755	5.591	4.214	5.882	5.199	8.790	9.106	4.228	4.747
w/o DR	3.398	2.633	7.458	5.754	6.190	5.203	12.679	8.565	7.712	7.256

Note: * indicates statistically significant improvements over best baseline by an unpaired two-sample t-test with p-value = 0.01.

Multi-step Refocusing. Table 6 shows the effect given different numbers of refocus steps g_r. UCPR achieves better performance with a large number of refocus steps, suggesting that this enables the generation of refined user portfolio information. Also, in Fig. 5(a), as the reasoning path moves from *Horns* to *Gothic fiction* due to the relation of book *Horns* to genre *Gothic fiction*, greater attention is put on entities such as *Fantasy novel* and *Amsterdam* after refocusing. This shows that the proposed multi-step refocusing successfully distinguishes the importance of relevant entities in the portfolio set, and salient entities guide the model to make the next movement with more persuasive evidence collected from the user perspective. A similar phenomenon can be observed from the instance in Fig. 5(b).

Dynamic Representation. The proposed dynamic representation allows UCPR to dynamically change the perspective of entities from portfolio set $\mathcal{P}_u$ according to the chosen entity. As shown in Table 9, removing the dynamic representation (UCPR$_{\text{w/o DR}}$) deteriorates performance, attesting its significance with respect to the proposed method. For simplicity, we employ only a linear-transformation-based method; future work would involve exploring updating methods which account for item-dependency relations such as Kang et al. [15], Wang et al. [26, 27].

6 USER STUDY

In addition to evaluating model performance and diversity using automatic metrics, we also conducted two human evaluations to investigate from the user's perspective whether 1) the generated reasoning path reasonably agrees with the corresponding highlighted entities, and 2) the guidance is informative for downstream users and systems to find the next potential interest products.

For the first experiment, annotators judged the relevance between the highlighted entities at each step and the UCPR-generated reasoning paths. *Relevance* was defined for the respondents. For example, we consider it relevant that in Fig. 4(b), at time step $t = 2$ and $t = 3$, the highlighted entities *fruits* and *Eau De Parfum Spray* are relevant to the path action *fruits* and *Eau De Parfum EDP*, respectively. The rating criteria is this: +2: The generated reasoning path matches the highlighted entities from the user portfolio for more than two steps. +1: Only one step from the generated reasoning path matches the highlighted entities; 0: The highlighted entities are irrelevant to the reasoning path. For comparison, we generated a *baseline for experiment 1* by matching highlighted entities from the user portfolio with randomly sampled reasoning paths.

For the second experiment, to asses whether highlighting entity guidance is informative for finding answers, respondents chose next-step entities based on the highlighted entities for three rounds to reach the final answer. Taking the path in Fig. 4(b) as an example, at the first round, time step $t = 1$ respondents were provided with starting entities *user 12661* and the model highlighted entities *Eau De Toilette Spray*, *Cacharel perfume*, and *user 27301* and were asked to choose the most relevant next-action entity from all possible actions, such as *Cacharel perfume*, *Britney Spears perfume*, and *Queen Helene Hand*. Then, at the next round, respondents were asked to choose the next-action entities based on the chosen entity at the previous round and the corresponding highlighted entities, similar to the final round. Each question was marked as 1 if the final round entity matched any of the user's interacted items; otherwise, it was marked as 0. For comparison, we generate a *baseline for experiment 2* by hiding the user portfolio and asked respondents to choose the next entities based on only the current step's entity information. For the two experiments, we recruited a total of 20 participants and administered an online quiz consisting of fifteen rounds. Each round was sampled from the test set of the corresponding dataset.

Table 10: Average score on two user study experiments. Larger values indicate better results.

Dataset	Clothing		Cell Phones		Beauty	
Metrics	Exp1	Exp2	Exp1	Exp2	Exp1	exp2
UCPR	1.34	7.7%	1.49	11.3%	1.32	12.7%
Baseline	0.69	1.4%	0.71	5.7%	0.85	6.0%

Table 10 summarizes the results. In Exp 1, in contrast to the baseline methods, respondents agreed more with the reasoning paths that contained matched highlighted entities, indicating that the highlighted entities reasonably matched the reasoning paths. In Exp 2, compared with the baseline, we find that overall accuracy was higher, showing that the highlighted entities were informative for humans to find products of potential interest to the user. Both results support the better explainability of our proposed method.

7 CONCLUSION

We propose UCPR, a user-centric path reasoning network in which a well-designed user demand portfolio guides the reasoning process to facilitate precise and explainable recommendations. Results show that the proposed user demand portfolio successfully guides the reasoning process in the user-centric view. In addition, multi-step refocusing and dynamic representation work well together, enabling sophisticated reasoning based on the user demand portfolio. The effectiveness of each proposed component is confirmed by an ablation study and a case study, which help to explain the superiority of UCPR. In the future, we will investigate leveraging the learned user demand portfolio and reasoning paths for downstream interactive product promotion.

ACKNOWLEDGEMENT

This research is partially supported by Ministry of Science and Technology, Taiwan under the project contract 109-2221-E-001-015- and 108-2221-E-001-012-MY3.

REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation. *CoRR* abs/1805.03352 (2018). arXiv:1805.03352 http://arxiv.org/abs/1805.03352

[2] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural News Recommendation with Long- and Short-term User Representations. https://doi.org/10.18653/v1/P19-1033

[3] Yang Bai, Ziran Li, and Hai-Tao Ding, Ning. [n.d.]. Infobox-to-text Generation with Tree-like Planning based Attention Network. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 2787–2795.

[5] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences. *CoRR* abs/1902.06236 (2019). arXiv:1902.06236 http://arxiv.org/abs/1902.06236

[6] Hongxu Chen, Yicong Li, Xiangguo Sun, Guandong Xu, and Hongzhi Yin. 2021. Temporal Meta-path Guided Explainable Recommendation. arXiv:2101.01433 [cs.SI]

[7] Xu Chen, Yongfeng Zhang, and Zheng Qin. 2019. Dynamic Explainable Recommendation Based on Neural Attentive Models. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 53–60. https://doi.org/10.1609/aaai.v33i01.330153

[8] Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. 2019. UHop: An Unrestricted-Hop Relation Extraction Framework for Knowledge-Based Question Answering. *CoRR* abs/1904.01246 (2019). arXiv:1904.01246 http://arxiv.org/abs/1904.01246

[9] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alexander J. Smola, and Andrew McCallum. 2018. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. *CoRR* abs/1711.05851 (2018). arXiv:1711.05851 http://arxiv.org/abs/1711.05851

[10] Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, and Gerard de Melo. 2020. Fairness-Aware Explainable Recommendation over Knowledge Graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 69–78. https://doi.org/10.1145/3397271.3401051

[11] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. *CoRR* abs/1603.06393 (2016). arXiv:1603.06393 http://arxiv.org/abs/1603.06393

[12] Ruining He and Julian J. McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. *CoRR* abs/1602.01585 (2016). arXiv:1602.01585 http://arxiv.org/abs/1602.01585

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[14] Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2019. Graph Neural News Recommendation with Long-term and Short-term Interest Modeling. *arXiv e-prints*, Article arXiv:1910.14025 (Oct. 2019), arXiv:1910.14025 pages. arXiv:1910.14025 [cs.IR]

[15] Wang-Cheng Kang, Mengting Wan, and Julian J. McAuley. 2018. Recommendation Through Mixtures of Heterogeneous Item Relationships. *CoRR* abs/1808.10031 (2018). arXiv:1808.10031 http://arxiv.org/abs/1808.10031

[16] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. 1831–1839. https://doi.org/10.1145/3219819.3219950

[17] Kechen Qin, Yu Wang, Cheng Li, Kalpa Gunaratna, Hongxia Jin, Virgil Pavlu, and Javed A. Aslam. 2020. A Complex KBQA System using Multiple Reasoning Paths. arXiv:2005.10970 [cs.LG]

[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *CoRR* abs/1205.2618 (2012). arXiv:1205.2618 http://arxiv.org/abs/1205.2618

[19] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. *CoRR* abs/1704.04368 (2017). arXiv:1704.04368 http://arxiv.org/abs/1704.04368

[20] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2016. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues. *CoRR* abs/1605.06069. arXiv:1605.06069 http://arxiv.org/abs/1605.06069

[21] Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and Diverse Text Generation with Planning-based Hierarchical Variational Model. *arXiv e-prints*, Article arXiv:1908.06605 (Aug. 2019), arXiv:1908.06605 pages. arXiv:1908.06605 [cs.CL]

[22] Dinghan Shen, Asli Çelikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin. 2019. Towards Generating Long and Coherent Text with Multi-Level Latent Variable Models. *CoRR* abs/1902.00154 (2019). arXiv:1902.00154 http://arxiv.org/abs/1902.00154

[23] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent Knowledge Graph Embedding for Effective Recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 297–305. https://doi.org/10.1145/3240323.3240361

[24] Chang-You Tai, Meng-Ru Wu, Yun-Wei Chu, Shao-Yu Chu, and Lun-Wei Ku. 2020. MVIN: Learning Multiview Items for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 99–108. https://doi.org/10.1145/3397271.3401126

[25] N. Tintarev and J. Masthoff. 2007. A Survey of Explanations in Recommender Systems. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*. 801–810.

[26] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling Item-Specific Temporal Dynamics of Repeat Consumption for Recommender Systems. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 1977–1987. https://doi.org/10.1145/3308558.3313594

[27] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make It a Chorus: Knowledge- and Time-Aware Item Modeling for Sequential Recommendation. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3397271.3401131

[28] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripple Network: Propagating User Preferences on the Knowledge Graph for Recommender Systems. *CoRR* abs/1803.03467 (2018). arXiv:1803.03467 http://arxiv.org/abs/1803.03467

[29] 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* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1835–1844. https://doi.org/10.1145/3178876.3186175

[30] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge Graph Convolutional Networks for Recommender Systems. *CoRR* abs/1904.12575 (2019). arXiv:1904.12575 http://arxiv.org/abs/1904.12575

[31] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. *CoRR* abs/1905.07854 (2019). arXiv:1905.07854 http://arxiv.org/abs/1905.07854

[32] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2018. Explainable Reasoning over Knowledge Graphs for Recommendation. *CoRR* abs/1811.04540 (2018). arXiv:1811.04540 http://arxiv.org/abs/1811.04540

[33] Zhen Wang, J. Zhang, Jianlin Feng, and Z. Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*.

[34] Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. *CoRR* abs/1906.05237 (2019). arXiv:1906.05237 http://arxiv.org/abs/1906.05237

[35] Yikun Xian, Zuohui Fu, Handong Zhao, Yingqiang Ge, Xu Chen, Qiaoying Huang, Shijie Geng, Zhou Qin, Gerard de Melo, S. Muthukrishnan, and et al. 2020. Cafe: Coarse-to-Fine Neural Symbolic Reasoning for Explainable Recommendation. *Proceedings of the 29th ACM International Conference on Information Knowledge Management* (Oct 2020). https://doi.org/10.1145/3340531.3412038

[36] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2016. Topic Augmented Neural Response Generation with a Joint Attention Mechanism. *CoRR* abs/1606.08340 (2016). arXiv:1606.08340 http://arxiv.org/abs/1606.08340

[37] Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. *CoRR* abs/1707.06690 (2017). arXiv:1707.06690 http://arxiv.org/abs/1707.06690

[38] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized Entity Recommendation: A Heterogeneous Information Network Approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining* (New York, New York, USA) *(WSDM '14)*. ACM, New York, NY, USA, 283–292. https://doi.org/10.1145/2556195.2556259

[39] 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 (KDD '16)*. 353–362. https://doi.org/10.1145/2939672.2939673

[40] 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* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 353–362. https://doi.org/10.1145/2939672.2939673

[41] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W. Bruce Croft. 2017. Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 1449–1458. https://doi.org/10.1145/3132847.3132892

[42] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD '17)*. ACM, New York, NY, USA, 635–644. https://doi.org/10.1145/3097983.3098063

[43] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. 2020. Leveraging Demonstrations for Reinforcement Recommendation Reasoning over Knowledge Graphs. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3397271.3401171

[44] Wayne Xin Zhao, Gaole He, Hong-Jian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2018. KB4Rec: A Dataset for Linking Knowledge Bases with Recommender Systems. *CoRR* abs/1807.11141 (2018). arXiv:1807.11141 http://arxiv.org/abs/1807.11141

[45] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. *CoRR* abs/1701.04783 (2017). arXiv:1701.04783 http://arxiv.org/abs/1701.04783

[46] Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. 2018. An Interpretable Reasoning Network for Multi-Relation Question Answering. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2010–2022. https://www.aclweb.org/anthology/C18-1171