Query-Aware Explainable Product Search With Reinforcement Knowledge Graph Reasoning

Qiannan Zhu, Haobo Zhang, Qing He, and Zhicheng Dou, *Member, IEEE*

***Abstract*—Product search is one of the most effective tools for people to browse and purchase products on e-commerce platforms. Recent advances have mainly focused on ranking products by their likelihood to be purchased through retrieval models. However, they overlook the problem that users may not understand why certain products are retrieved for them. The lack of appropriate explanations can lead to an unsatisfactory user experience and further decrease user trust in the platforms. To address this problem, we propose a Query-aware Explainable Product Search with Reinforcement Knowledge Reasoning, namely *QEPS*, which uses search behaviors related to the current query to reinforce explanations. Specifically, with the aim of retrieving suitable products with explanations, QEPS takes full advantage of the user-product knowledge graph (KG) and develops a reinforcement learning approach, characterized by the demonstration-guided policy network and query-aware rewards, to perform explicit multi-step reasoning on the KG. The reasoning paths between users and products are automatically derived from the current query-related search behavior, which can provide valuable signals as to why the retrieved products are more likely to satisfy the user's search intent. Empirical experiments on four datasets show that our model achieves remarkable performance and is able to generate reasonable explanations for the search results.**

***Index Terms*—Explainability, knowledge reasoning, product search, reinforcement learning.**

I. Introduction

With the rapid growth of products on e-commerce websites, users are inundated with a vast array of choices and options in their daily lives. Product search engines have become an increasingly popular tool for people to discover and purchase products. In a typical product search scenario, a user submits a query to a search engine and the search engine returns a list of relevant products ranked by likelihood of purchase. The relevance of the products on the search engine results pages would affect user satisfaction and transactions, which in turn would affect the revenues and profits of the e-commerce platforms.

Due to the important influence of the user's personal preferences on their purchase decision, previous studies on product search such as HEM [1], TEM [2] and ZAM [3] attempt to incorporate personalization to improve the quality of product search, aiming at retrieving relevant products to satisfy the user's personal tastes and preferences. For example, the promising model ZAM [3] designs a zero-attention mechanism on the user's search logs to automatically identify the user's individual intent on the product search session. However, such personalized product search methods ignore the problem that users may not understand why certain products are returned to them. A good reason makes the search results more trustworthy and is essential for users to justify their purchases. Therefore, it is important to develop an effective product search model with the ability to retrieve relevant products and provide good explanations for the retrieved results.

Based on the successful application of knowledge graphs (KGs) in explainable product recommendations [4], [5], [6], [7], [8], it is reasonable to assume that structured information in KGs has great potential in providing explanations for product search. Inspired by this assumption, recent methods such as DREM [9] attempt to represent the queries as dynamic relationships between users and products in the user-product KG, and employ entity soft matching with knowledge embeddings to extract the ad-hoc explanations in the form of reasoning paths. Despite their effectiveness, these explainable product search methods suffer from two inherent limitations. First, they simply adopt the entity soft matching strategy to find the soft entities in the reasoning paths between users and products, preferring to generate implicit explanations. Second, they ignore the potential of search behaviors related to the current query on the generation of explanations, resulting in the under-exploration of the collaborative signal related to the current query. Therefore, these limitations act as obstacles to providing explicit and persuasive explanations tailored to the current preferences of the users.

To address the above issues, this paper *considers the search behaviors related to the current query as the direct way to enhance the explicit and persuasive explanations*. More importantly, the search behaviors related to the current query are the queries that have semantic relevance (similarity) to the current query, which can be considered as the collaborative information of the current query. The typical knowledge-aware

Manuscript received 26 December 2022; revised 22 May 2023; accepted 11 July 2023. Date of publication 24 July 2023; date of current version 6 February 2024. This work was supported in part by the National Key R&D Program of China under Grant 2022ZD0120103, in part by the National Natural Science Foundation of China under Grants 62102421 and 62272467, in part by Beijing Outstanding Young Scientist Program under Grant BJJWZYJH012019100020098, and in part by the Public Computing Cloud, Renmin University of China, and the fund for building world-class universities (disciplines) of Renmin University of China. Recommended for acceptance by Z. Guan. *(Corresponding author: Zhicheng Dou.)*

Qiannan Zhu is with the School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China (e-mail: zhuqiannan@bnu.edu.cn).

Haobo Zhang and Zhicheng Dou are with the Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China (e-mail: 2018200680@ruc.edu.cn; dou@ruc.edu.cn).

Qing He is with the School of Finance, Renmin University of China, Beijing 100872, China (e-mail: 2019200172@ruc.edu.cn).

Digital Object Identifier 10.1109/TKDE.2023.3297331

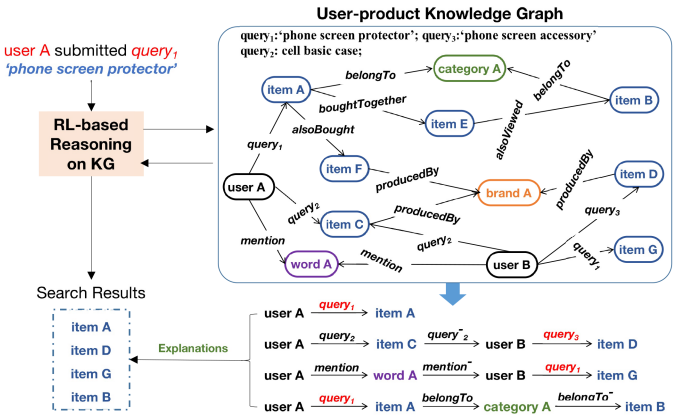

Fig. 1. Illustration of Knowledge Reasoning for Explainable Product Search. The path-based explanations take into account the search behaviors $\{query_1, query_3\}$ related to the current query $query_1$. The query-related search behaviors are the key to guaranteeing consistency between the explanation and the users' current search intentions.

explanations [6], [10] in recommendation are multi-hop path reasoning, which focuses on entity and relation selection during the reasoning process. Similarly, in the product search scenario, the path-based explanation that implies the search behaviors related to the current query will be more convincing and well-understood. Fig. 1 illustrates an example where user A entered *query*$_1$, the path (user A, *query*$_1$, item A) can be explained as *item A is retrieved because the user previously purchased this product under the submitted query*$_1$, and the path (user A, *query*$_2$, item C) $\rightarrow$ (item C, *query*$_2$, user B) $\Rightarrow$ (user B, *query*$_1$, item D) explains that *item D was retrieved because user B, with similar search behavior to target user A, had previously purchased this product under the submitted query*$_1$. The relevant search behaviors *query*$_1$ in the paths are the collaborative signal about the current query *query*$_1$, which is the key to guaranteeing consistency between the explanation and the users' current search preference.

In order to fully exploit the current query and explore the reasoning paths consistent with it, we propose a query-aware explainable product search model with reinforcement knowledge reasoning, named QEPS. Different from the typical reinforcement learning-based recommendation methods [6], [10], QEPS considers the current query as guidance to explore its relevant search behaviors in the path-finding process, encouraging the agent to arrive at the appropriate products as well as explanations tailored to the user's current preferences. Specifically, QEPS develops the demonstration-guided actor-critic framework, which is guided by the query-specific path demonstrations and query-aware reward signals to perform path reasoning and return the explainable search results to the users. For the query-specific path demonstrations, QEPS designs a demonstration extractor to heuristically generate the query-specific demonstrations that imply query-related search behaviors based on manually defined meta-paths. These demonstrations are the typical ground truth of the path-based explanations for the current query, which can provide supervised guidance to the actor-critic network's path-finding process. For the query-aware reward signals, QEPS determines the terminal and immediate rewards, which comprehensively consider the plausibility of the triplet with the current query (user, *query*, item), the rationalization of reasoning paths supervised by the query-specific path demonstrations and the semantic relevance between current query and the retrieved products. Such rewards can well judge whether the agent generates a demonstration-like reasoning path and whether the retrieved product satisfies the user's current preference. We empirically evaluate our model on Amazon e-commerce datasets. Experimental results show that our model achieves remarkable retrieval results as well as reasonable explanations.

The main contributions of this paper can be outlined as follows.

- We propose an RL-based approach to perform an explainable product search, which considers the query-related search behavior as guidance to explore a set of products with explanations to satisfy users' search intent.
- We design a demonstration-guided policy network characterized by path demonstrations and query-aware rewards to efficiently generate the explicit reasoning paths that are towards suitable items tailored to the user's current preferences.
- We develop query-aware rewards that leverage the rationalization of the query-related triplets and reasoning paths together with the relevance of the retrieved products to the user's search intent to ensure the effectiveness of explanations.

II. Preliminary

In an e-commerce search scenario, a user-product KG G_s is constructed from the product-related meta-information and user-product interactions. Typically, the user-product KG is composed of triplets, i.e., $G_s = \{(e, r, e')|e, e' \in E, r \in R\}$, where E and R are the sets of entity and relation, respectively. Each triplet (e, r, e') represents the fact that the relation r links the head entity e and the tail entity e'. Following DREM [9], we convert five types of entities (i.e., *user, item, word, brand, category*) and their relations into the following groups of triplet facts: (1) (u, q, i) represents that the user u bought the item i under the submitted query q. (2) *(u, mentioned, w)* represents that the user u mentioned the word w in his reviews. (3) *(i, describedAs, w)* represents that the item i was described by the word w in the item's reviews. (4) *(i, belongTo, c)* represents that the item i belongs to the category c. (5) *(i, producedBy, b)* represents that the item i is produced by the brand b. (6) *(i_1, alsoBought, i_2)* represents that the items i_1 and i_2 were bought by the same user. (7) *(i_1, boughtTogether, i_2)* represents that the items i_1 and i_2 were bought together in a single transaction. (8) *(i_1, alsoViewed, i_2)* represents that the item i_2 was viewed before or after the purchase of the item i_1. The Table I introduces the basic notations that will be used in this paper.

Due to the nature of search tasks, the queries can be considered as dynamic relationships between users and items because they are usually computed by the search content for product search on the fly. Previous studies show that the non-linear projection function [11] is more robust to modeling queries in latent space.

TABLE I
SUMMARY OF THE NOTATIONS USED IN THIS PAPER

Symbol	Description
G_s	the user-product knowledge graph (KG)
E	the entity set of the user-product KG
R	the relation set of the user-product KG
(h, r, t)	the triplet in the user-product KG
I_u	the item set purchased by the user u
M	the meta-path set
q	the user's current query
w	the word in reviews or queries from users (items)
d	the dimension of the embeddings
s_t	the state at t-th step
$\tilde{A}_t, A_t$	the (pruned) action space at t-th step
$R_{u,e}$	the knowledge-based terminal reward
$R_{q,q'}$	the path-based terminal reward
$R_{q,e}$	the semantic-based terminal reward
$R_{m,t}$	the immediate reward
$\lambda_1, \lambda_2, \lambda_3, \lambda_4$	the weights of the rewards
$\pi_\theta, \pi_\varphi, \pi_\phi$	the actor, demonstration-based indicator and critic
$\mathbf{h}, \mathbf{t}$	the embedding of the head (tail) entity $h(t)$
$\mathbf{r}$	the embedding of the relation r
$\mathbf{q}$	the embedding of the query q
$\mathbf{w}$	the embedding of the word w

Let $\{\mathbf{w}_i \in R^d\}$ be the word embeddings of words $\{w_i\}$ in the query q, we calculate the representation of the query q as

$$\mathbf{q} = \tanh\left(\mathbf{U}\frac{\sum_{w_i \in q} \mathbf{w}_i}{|q|} + \mathbf{b}\right) \tag{1}$$

where $\mathbf{U} \in R^{d\times d}, \mathbf{b} \in R^d$ are the parameters and d is the dimension of the embeddings of entities and relations in the user-product KG.

Task Problem: The problem in this paper is to find product items that are likely to have a *query* relationship with users, i.e., to measure the plausibility of the triplet (u, q, i). In particular, given a user-product KG G_s, the user u and his current query q, the goal is to retrieve a set of product items $\{i_n\} \in E$ such that each triplet (u, q, i_n) is associated with one reasoning path tailored to user's current query as $p(u, q, i_n) = \{u, r_1, e_1, \ldots, r_k, i_n\}, 1 \leqslant k \leqslant T$. T is the maximum length of the path.

III. METHODOLOGY

Our model QEPS aims to provide explicable search results through explicit KG reasoning. The main idea is to train an RL agent under the guidance of path demonstrations and query-aware rewards, where the agent is encouraged to find paths to potential items conditioned on the user and his current query. To achieve this, QEPS takes the current query as the guidance signal in the path-finding process and designs three main components: (1) Demonstration Extractor (DE) : DE heuristically extracts the query-specific path demonstrations that imply query-related search behaviors based on manually defined meta-paths. These query-specific demonstrations can provide supervised guidance to the actor-critic network for retrieving suitable products and explanations. (2) Query-aware Rewards (QR): QR thoroughly considers the plausibility of the query-related triplet (u, q, i) and the reasoning path together with the semantic relevance between the current query and the retrieved product, and further elaborates the terminal and immediate rewards to assess the rationalization of the path-finding process. (3) Demonstration-guided Policy (DP): DP is an actor-critic network where the goal is to perform path reasoning on the selection of appropriate entity-relation pairs to finally reach the correct product. The policy network is jointly supervised and optimized by the query-specific path demonstrations and query-aware reward signals. The framework of our model QEPS is illustrated in Fig. 2.

A. Reinforcement Learning Formulation

Starting from a user node, QEPS sequentially determines the next-hop nodes and moves towards potential items that satisfy the user's search intent (i.e., the current query). To achieve it, we formulate the knowledge reasoning as a Markov Decision Process (MDP) and define the following components:

State: The state s_t at step t is defined as $s_t = (u, q, h_t, e_t)$, where u is the starting user entity, q is the current query, h_t is the history before step t, i.e., $h_t = \{u, r_1, e_1, \ldots, e_{t-1}, r_t\}$, e_t is the entity visited at step t. (u, q) is the global context shared by all states. Conditioned on the user u and his current query q, the initial state is $s_0 = (u, q, u, \oslash)$ and the terminal state is $s_T = (u, q, h_T, e_T)$.

Action: The possible action space A_t of state s_t is defined as the outgoing edges of entity e_t in G_s excluding history entities. Concretely, $A_t = \{(r', e')|(e_t, r', e') \in G_s, e' \notin \{u, e_1, \ldots, e_t\}\}$. To end the search of the agent, a self-loop edge associated with the no operation (NO-OP) relation is added to each entity, i.e., if $e \in E$, then $(e, r_{noop}, e) \in G_s$. We also add reverse edges to guarantee the path connectivity in the reasoning process, i.e., if $(e, r, e') \in G_s$, then $(e', r^-, e) \in G_s$. Since the out-degrees of some entities can be very large, it is flexible to keep the size of the action space according to their out-degree. Thus we keep the most promising edges with a pruning function as

$$f((r', e')|\, u, h_t) = \left(\mathbf{u} + \sum_{r_i \in h_t} \mathbf{r}_i + \mathbf{r}'\right) \odot \mathbf{e}' \tag{2}$$

where $\odot$ is the dot product operation. The pruning function is to map the outgoing edge (r', e') to a real-valued score for the action selection. Importantly, the score can measure the likelihood that a user u, a sequence of relations $\{r_1, \ldots, r_t, r'\}$ and an entity e' can form a reasonable path during the action selection. Then the pruned action space of state s_t, denoted by $\tilde{A}_t(u)$, is defined as

$$\tilde{A}_t(u) = \{(r', e')|\mathrm{Top}_\epsilon(\mathrm{rank}(f)), (r', e') \in A_t\} \tag{3}$$

where $\mathrm{rank}()$ is the ranking function in descending order, $\mathrm{Top}_\epsilon()$ is to select the top ϵ actions according to their ranking scores in the (2).

Transition: Given a state $s_t = (u, q, h_t, e_t)$ and an action $a_t = (r_{t+1}, e_{t+1})$, the transition to the next state is $s_{t+1} = (u, q, h_{t+1}, e_{t+1})$, where $h_{t+1} = \{u, r_1, e_1, \ldots, r_t, e_t, r_{t+1}\}$.

Reward: The agent aims to explore as many valuable paths as possible that fit the user's search intent. A valuable path is the one that terminates with a high probability at an item relevant to

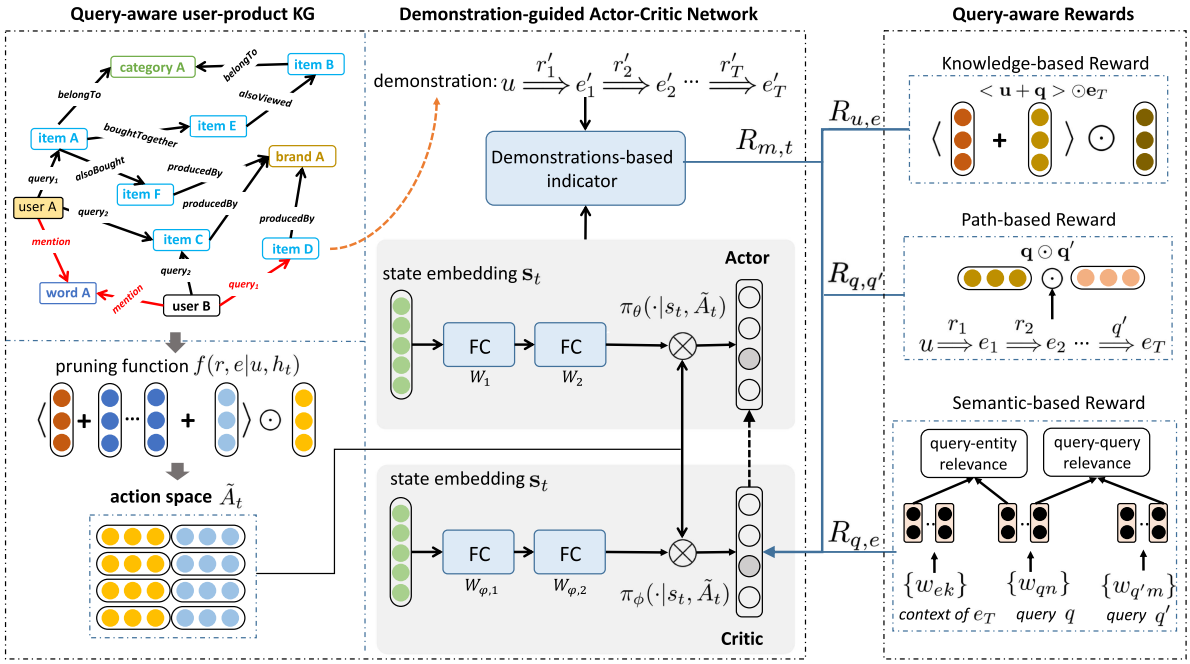

Fig. 2. Our Query-aware Reinforcement Knowledge Graph Reasoning Model QEPS for Explainable Product Search. FC stands for the fully connected layer.

the user's current. We define the reward function for the terminal state $s_T = (u, q, h_T, e_T)$ as

$$R_T = \begin{cases} g(u, q, e_T), & \text{if } e_T \in I_u \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where I_u is the set of items purchased by the user u, $g(\cdot)$ is an aggregated reward function that motivates the agent to find both correct items and explanations. The details of the reward function are described in Section III-C.

B. Demonstration Extractor

The demonstration extractor aims to obtain a set of user-centric path demonstrations, which are the query-aware multi-hop paths between users and items in the user-product KG. Studies suggest [12], [13] that the meta-paths tend to export the path demonstrations with more interpretable and logical than randomly sampled paths. To generate the high-quality path demonstrations, we define several typical meta-paths followed by [6], [10], [14] in Table II.

As illustrated in Table II, a meta-path is a sequence of entity and relation types [12]. For example, the meta-path of the path *Jark* $\xrightarrow{q:\{iphone\}}$ *iphone6* $\xrightarrow{belongTo}$ *Apple phone* $\xrightarrow{belongTo^-}$ *iphone12* can be written as *user* $\xrightarrow{q:\{w_i\}}$ *item* $\xrightarrow{belongTo}$ *category* $\xrightarrow{belongTo^-}$ *item*. In the demonstration extractor, we first consider each user u as the starting node of the constrained random walks [10] and sample only the paths whose meta-path belongs to the predefined set. Second, we keep only those paths that lead to the items purchased by the user as demonstrations. Formally, we assume that the pre-defined metapath set is $\{M_i|i \in [1, m]\}$ and the kept path demonstrations of user u are $P_u = \{P_i|P_i = \{p_{ij}\}\}$. m is the number of the metapaths and P_i is the set of path demonstrations belonging to the metapath M_i.

TABLE II
THE PRE-DEFINED META-PATHS IN THIS PAPER

(1)	user $\xrightarrow{q:\{w_i\}}$ item
(2)	user $\xrightarrow{q:\{w_i\}}$ item $\xrightarrow{q^-:\{w_i\}}$ user $\xrightarrow{q:\{w_i\}}$ item
(3)	user $\xrightarrow{mention}$ word $\xrightarrow{mention^-}$ user $\xrightarrow{q:\{w_i\}}$ item
(4)	user $\xrightarrow{q:\{w_i\}}$ item $\xrightarrow{belongTo}$ category $\xrightarrow{belongTo^-}$ item
(5)	user $\xrightarrow{q:\{w_i\}}$ item $\xrightarrow{producedBy}$ brand $\xrightarrow{producedBy^-}$ item
(6)	user $\xrightarrow{q:\{w_i\}}$ item $\xrightarrow{describedAs}$ word $\xrightarrow{describedAs^-}$ item
(7)	user $\xrightarrow{q:\{w_i\}}$ item $\xrightarrow{relatedRelation}$ item $\xrightarrow{relatedRelation^-}$ item

$q : \{w_i\}$ is the query relation q with its search content $\{w_i\}$. Relatedrelation is alsobought, alsoviewed or boughttogether relation.

In order to obtain the ground truth of the path-based explanations for the current query, i.e., the path demonstrations as the examples in Fig. 1 that contains query-related search behaviors, we directly use the relevance of the path demonstrations to the current query to determine how the demonstration can explain the search results. Before doing so, we introduce the concept of *desirable-query* as follows.

Definition 1(desirable-query): For the relation sequence $\{r_1, r_2, \ldots, r_j, r_{j+1} \ldots, r_k\}$ in a path demonstration $\{u, r_1, e_1, \ldots, r_j, e_j, \ldots, r_k, e_k\}$, if there exists a *query* relation r_j whose following relations $\{r_{j+1}, \ldots, r_k\}$ are not *query* relation, then r_j is the *desirable-query*.

Based on the characteristics of the meta-paths and the *desirable-query*, a naive way is to use the similarity or relevance of the current query to the *desirable-query* to explain why the

retrieved items match the user's search intents. The stronger the similarity, the better the path-based explanation matches the user's search intents. For example, given the user u and his current query *Apple Phone*, and two path demonstrations *Jenny* $\xrightarrow{mention}$ *high-resolution* $\xrightarrow{mention^-}$ *Bob* $\xrightarrow{q:\{iPhones\}}$ *iPhone12 Plus* and *Jenny* $\xrightarrow{mention}$ *high-resolution* $\xrightarrow{mention^-}$ *Bob* $\xrightarrow{q:\{CellPhones\ Protector\}}$ *Screen Protector for iPhone12 Plus*. The first path returns a more appropriate item *iPhone12 Plus* and a more convincing explanation than the second one. This is because the *desirable-query {iPhones}* in the first path is more similar or relevant to the current query *Apple Phone* than the *desirable-query {CellPhones Protector}* in the second path. Thus, the *desirable-query* with the high similarity to the current query are the users' relevant search behaviors, which are the key to ensuring the consistency between the explanation and the users' current preference.

Having established the demonstrations with stronger explanations for the current query, we select the path demonstrations from $\{P_i\}$ with a ratio α by ranking the similarity of the current query and the *desirable-query* in descending order, i.e,

$$P_u = \{\text{rank}(\text{sim}(\mathbf{q}', \mathbf{q})) \leq \alpha | q' \in p_{ij}, p_{ij} \in P_i, i \in [1, m]\} \tag{5}$$

where $\mathbf{q}$ and $\mathbf{q}'$ are the embeddings of the current query q and the *desirable-query* q' in the path demonstration p_{ij}. *sim(·)* is the cosine function, α is a hyperparameter that upper bounds the size of the path demonstrations. After that, the selected demonstrations are the paths that contain query-related search behaviors. In this paper, the embeddings of queries are calculated by (1) and pre-trained by the typical method DREM [9].

C. Query-Aware Rewards

Better reward design can reflect the uncertainty of how the product item satisfies the user's search intent. This section comprehensively considers the plausibility of the triplet with the current query (u, q, i), the explainability of the query-aware reasoning path, and the semantic relevance of the current query to the arrived product item, and further defines three terminal rewards to estimate the correctness of the reached entities.

1) Knowledge-Based Reward: From the perspective of knowledge graphs, the product search in this paper aims to find product items that are likely to have the *query* relationship with users. As studied in the knowledge representation learning methods [15], [16], [17], [18], [19], the observed triplets in KGs should have higher plausibility than these unobserved triplets. For example, the typical knowledge embedding method TransE [15] makes the observed triplet (h, r, t) satisfy $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, but forces $\mathbf{h} + \mathbf{r}$ away from $\mathbf{t}'$ for the unobserved triplet (h, r, t'). Thus, we use the plausibility of the query triplet (u, q, e_T) to estimate the probability of the user u purchasing the product item i based on the query q

$$R_{u,e} = (\mathbf{u} + \mathbf{q}) \odot \mathbf{e}_T \tag{6}$$

where $\{\mathbf{u}, \mathbf{q}, \mathbf{e}_T\} \in R^d$ are the embeddings of the user u, the query q and the reached terminal item e_T respectively. Since learning the embeddings of entities and relations in the user-product KG is not our focus in this paper, we simply pre-trained these embeddings using DREM [9] with the goal of fully exploiting the structural information of KGs. These pre-trained embeddings will also be used in our subsequent rewards.

2) Path-Based Reward: The reasoning paths in our model aim to explain to users why the terminal items are retrieved. A better reasoning path can provide a more convincing explanation to fit the user's search intentions. After establishing the *desirable-query*, if the *desirable-query* in the reasoning path is relevant or similar to the current query, then the path can infer the correct target item and generate a better explanation. For example, given the user u with his current query q, a path belonging to the meta-path $u \xrightarrow{mention} word \xrightarrow{mention^-} u' \xrightarrow{q'} i$, if the *desirable-query* q' is relevant or similar to the current query q, then the agent can return the item i and give the explanation as *the item i is retrieved because the users with similar tastes to the target user u have previously purchased this product under the similar query q'*. Thus, we use the cosine similarity of the *desirable-query* q' in the reasoning path and the current query q as a reward to measure the explainability of the reasoning path, i.e,

$$R_{q,q'} = \text{sigmoid}(\cos(\mathbf{q}, \mathbf{q}')) \tag{7}$$

where $\{\mathbf{q}, \mathbf{q}'\}$ are the embeddings of the queries $\{q, q'\}$ pre-trained by DREM [9]. In summary, the path-based reward measures the explainability of a reasoning path by whether the path contains the collaborative information (i.e., similar or relevant search behaviors) for the current query.

3) Semantic-Based Reward: In the web search field, when a user submits a query, the search engine retrieves and ranks the relevant documents to satisfy the user's search intents. In particular, the semantic relevance of the query-document pair is the core metric to rank the document list, where the relevant query-document pairs would receive higher matching scores than the irrelevant ones. As studied in [3], in the product scenario, the user often specifies the item's name directly in the query string, where some keywords are relevant to the item's context (i.e., title, description). On this basis, contextual relevance is an effective way to recognize whether the terminal product items satisfy the user's search intents. Therefore, we consider the semantic relevance to strengthen the terminal reward. More specifically, for the terminal product item e, we concatenate the item's title and description to form its context as $C(e) = \{w_j\}$ and calculate the semantic relevance of the item's context to the current query q as

$$\begin{aligned} R_{q,e} &= \text{sigmoid}(\text{KNRM}(\mathbb{Q}, \mathbb{E})) \\ \mathbb{Q} &= \sum_{w_i \in q} \text{tf-idf}_i \ \boldsymbol{w}_i \\ \mathbb{E} &= \sum_{w_j \in C(e)} \text{tf-idf}_j \ \boldsymbol{w}_j \end{aligned} \tag{8}$$

There have been numerous methods to build the semantic relevance of the query-document pairs, such as Doc2Vec [20], KNRM [21], HRNN [22] and BERT [23], etc. Since relevance

modeling is not our focus in this paper, we simply adopt the popular KNRM [21] with 11 kernels to calculate the semantic relevance of the query-item pairs. Here we use word2vec [24] as the embeddings of the words $\{w_i, w_j\}$ in (8), and calculate TF-IDF weights of words by the set of query-item pairs, where the context of the items can be as documents. Here, word2vec [24] differs from the embeddings of the word entities in the user-product KG learned by DREM [9]. Simultaneously, we also use the semantic relevance to boost path-based reward $R_{q,q'}$ and rewrite it as

$$\begin{aligned} R_{q,q'} &= \text{sigmoid}(\cos(q, q') + \text{KNRM}(\mathbb{Q}, \mathbb{Q}')) \\ \mathbb{Q} &= \frac{\sum_{w_i \in q} \boldsymbol{w}_i}{|q|} \\ \mathbb{Q}' &= \frac{\sum_{w_j \in q'} \boldsymbol{w}_j}{|q'|} \end{aligned} \tag{9}$$

In order to provide both high-quality items and convincing explanations, we integrate the above rewards as the terminal reward, i.e,

$$R_T = \begin{cases} \lambda_1 R_{u,e} + \lambda_2 R_{q,e} + \lambda_3 R_{q,q'}, & \text{if } e_T \in I_u \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the weights of the three rewards. The terminal reward R_T is calculated when the agent terminates the search.

D. Demonstration-Guided Policy Network

This component is an actor-critic network that aims to retrieve a set of items for users, as well as the reasoning paths for the retrieved items. As described in Section III-B, the path demonstrations are a powerful tool that can arrive at correct items and provide convincing explanations. A straightforward way is to use the path demonstrations as ground-truth labels and learn a model that tries to generate paths identical to the demonstrations. To imitate the demonstrations, we design a demonstration-based indicator to supervise the actor's path generation based on the demonstrations, the actor's imitation learning can give the immediate reward $R_{m,t}$ to measure whether each step of the path generation is identical to the demonstrations. Moreover, the terminal reward is the direct signal to inform the actor whether the terminal entity is correct for the users. Thus, the critic jointly uses the terminal and immediate rewards together to accurately assess the value of each action with an unbiased estimate of the reward gradient for the actor's training.

1) Actor: In the t-th step, the actor π_θ takes as input the state s_t and the pruned action space $\tilde{A}_t(u)$, and emits the probability of each action in the action space as

$$\begin{aligned} \pi_\theta(\cdot|s_t, \tilde{A}_t) &= \text{softmax}(\tilde{\mathbf{A}}_t \odot g(s_t)) \\ g(s_t) &= \mathbf{W}_2 \text{ReLU}(\mathbf{W}_1\ \mathbf{s}_t) \end{aligned} \tag{11}$$

where $\mathbf{s}_t \in R^{d_s}$ is the concatenation of the embeddings of its elements, i.e, $\mathbf{s}_t = [\mathbf{u}; \mathbf{q}; \mathbf{h}_t; \mathbf{e}_t]$, $\tilde{\mathbf{A}}_\mathbf{t}$ is the stack of the embeddings $[\mathbf{r}; \mathbf{e}]$ of the actions $(r, e) \in \tilde{A}_t$, $\mathbf{W}_2 \in R^{2^{\tilde{}} d \times d_s}$, $\mathbf{W}_1 \in R^{d_s \times d_s}$ are the learned parameters in our model.

2) Demonstration-Based Indicator: The indicator judges whether the actor can generate a demonstration-like path segment at each step t. The path segment can be represented by s_t and the selected action a_t by the actor. Then the demonstration-based indicator π_φ is defined with the parameter φ

$$\begin{aligned} \pi_\varphi(s_t, a_t) &= \sigma(\rho_\varphi^T \tanh(\mathbf{W}_\varphi\ \overline{\mathbf{h}}_t) \\ \overline{\mathbf{h}}_t &= \tanh([\mathbf{s}_t; \mathbf{a}_{\varphi,t}]) \end{aligned} \tag{12}$$

where $\mathbf{a}_{\varphi,t} \in R^{d_a}$ is the learned embedding of action a_t in π_φ, σ is the sigmoid function and $\rho_\varphi \in R^{d_\varphi}, \mathbf{W}_\varphi \in R^{d_\varphi \times (d_s + d_a)}$ are key parameters to be learned. The indicator is trained so that $\pi_\varphi(s_t, a_t)$ can transmit the probability that (s_t, a_t) comes from a demonstration. This is achieved by the following classification loss, i.e,

$$L_\varphi = -(\log\ \pi_\varphi(s'_t, a'_t) + \log(1 - \pi_\varphi(s_t, a_t))) \tag{13}$$

where $s'_t = (u, q, h'_t, e'_t)$ and $a'_t = (r'_{t+1}, e'_{t+1})$ are determined by a demonstration $u \xrightarrow{r'_1} e'_1 \xrightarrow{r'_2} e'_2 \ldots \xrightarrow{r'_k} i$. Here $h'_t = \{u, r'_1, e'_1, \ldots, e'_{t-1}, r'_t\}$. In the experiment, we randomly sampled the demonstration from the demonstration set P_u. Then, the demonstration-based indicator rewards the actor if the actor generates (s_t, a_t) pairs that are likely to come from the demonstrations, the immediate reward $R_{m,t}$ is as follows:

$$R_{m,t} = \log\ \pi_\varphi(s_t, a_t) - \log(1 - \pi_\varphi(s_t, a_t)) \tag{14}$$

3) Critic: The critic is to effectively model the terminal and immediate rewards, which can accurately estimate the contribution of each action to the rewards for better guidance to the actor. To achieve this, we adopt the critic network [25] with better convergence properties to estimate the value (contribution) of each action. Specifically, given the state s_t and the action a_t of the t-th step, the critic network is defined as

$$\begin{aligned} \pi_\phi(s_t, a_t) &= (\mathbf{a}_{\phi,t} \odot g(s_t)) \\ g(s_t) &= \mathbf{W}_{\phi,2} \text{ReLU}(\mathbf{W}_{\phi,1}\ \mathbf{s}_t) \end{aligned} \tag{15}$$

where $\mathbf{a}_{\phi,t} \in R^{d_a}$, $\mathbf{W}_{\phi,2} \in R^{d_a \times d_s}, \mathbf{W}_{\phi,1} \in R^{d_s \times d_s}$ are the learned parameters in the critic network π_ϕ.

4) Optimization: We use the Temporal Difference (TD) method [26] to learn the critic network. This method first calculates the target q_t according to the Bellman equation[1]

$$q_t = R(t) + \mathbb{E}_{a \in \pi_\theta} \pi_\phi(s_{t+1}, a) \tag{16}$$

where $R(t)$ is an aggregated reward that motivates the policy to find paths similar to the demonstrations and to achieve better retrieval accuracy

$$R(t) = R_t + \lambda_4 R_{m,t},\ t \in [1, T] \tag{17}$$

Then the critic is updated by minimizing the TD error, i.e.

$$L_\phi = (\pi_\phi(s_t, a_t) - q_t)^2 \tag{18}$$

and the actor is learned by minimizing the loss function as

$$L_\theta = -\mathbb{E}_{a \in \pi_\theta} \pi_\phi(s_{t+1}, a) \tag{19}$$

[1] https://en.wikipedia.org/wiki/Bellman_equation

Algorithm 1: The Training Process of QEPS.

Input: the user u, the current query q, actor π_θ, critic π_ϕ, demonstration-based indicator π_φ, predefined demonstration path p' and the maximum path length T.
Output: π_θ, π_ϕ, π_φ.
for k = 1,... epoch **do**
  Initialize the total loss $L_\theta, L_\varphi, L_\phi = 0$, the reasoning path $P_0(u, i) = \{u\}$
  for t = 1,...,T **do**
    for $p \in P_{t-1}(u, i)$ **do**
      path $p = \{u, r_1, \ldots, r_{t-1}, e_{t-1}\}$
      state $s_{t-1} = (u, q, h_{t-1}, e_{t-1})$
      get pruned action set $\tilde{A}_{t-1}(u)$ using Eq (3)
      action probability $p(a) = \pi_\theta(a|s_{t-1}, \tilde{A}_{t-1}(u))$
      select an action $a = (r_t, e_t)$ based on the action probability distribution $\{p(a)|a \in \tilde{A}_{t-1}(u)\}$ randomly
      new path $P_t(u, i) = p \cup \{r_t, e_t\}$
      calculate R_t, q_t and the step loss $l_\theta, l_\varphi, l_\phi$ using p'
      new actor loss $L_\theta = L_\theta + l_\theta$
      new critic loss $L_\varphi = L_\varphi + l_\varphi$
      new indicator loss $L_\phi = L_\phi + l_\phi$
    end for
  end for
  successively minimize $L_\theta, L_\varphi, L_\phi$ to update π_θ, π_ϕ, π_φ.
end for
Return π_θ, π_ϕ, π_φ

We can jointly optimize the actor, critic, and demonstration-based indicator by minimizing the combined loss

$$L = L_\theta + L_\varphi + L_\phi \tag{20}$$

The process is described as Algorithm 1. It takes as input the given user u, the current query q, the predefined demonstration p', and the maximum path length T. As output, it delivers the optimized actor, critic and demonstration-based indicator by exploring the valuable reasoning path. In one iteration, the joint loss can be optimized by successively minimizing $L_\theta, L_\varphi, L_\phi$. More specifically, during the k-th iteration, for the pair of training sample (u, q), our agent samples a demonstration p_{ij} from P_u as well as a path p'_{ij} generated by the actor. Next, $L_\theta, L_\varphi, L_\phi$ are successively minimized based on p_{ij} and p'_{ij}. We then go to the next training epoch until the model converges or the maximum number of epochs is reached.

IV. Experiment

In this section, we conduct experiments with Amazon product datasets and compare our method with state-of-the-art baselines. In general, the evaluation of our model involves three parts: (1) the evaluation of retrieval performance in terms of retrieving items that the users are most likely to purchase under their submitted queries, (2) the evaluation of explanation effectiveness in terms of illustrating the case study of the connections between users, queries, and retrieved items, (3) the evaluation of the retrieval performance in terms of the relevant search behaviors and other hyper-parameters. The following presents the experimental setup, and then reports and analyzes the experimental results.

TABLE III
Statistics for the 5-Core Datasets of Amazon

	Electronics	Kindle	CD	CellPhones
#users	192,403	68,223	75,258	27,879
#items	63,001	61,934	64,443	10,429
#brands	3,525	1	1,414	955
#categories	983	2,523	770	206
#words	142,922	95,729	202,959	22,493
Train				
#queries	904	3313	534	134
#(user,query)	1,204,928	1,490,349	1,287,214	114,177
Test				
#queries	85	1290	160	31
#(user,query)	5,505	232,668	45,490	665

A. Experimental Setting

1) Data: Amazon product dataset[2] is one of the most popular and well-established benchmarks for product search [1], [3], [27] and recommendation [5], [6], [7]. It contains information for millions of customers, queries, products and associated metadata including descriptions, reviews, brands, and categories. In this paper, we use the 5-core data of four Amazon datasets, i.e., Electronics, Kinde Store, CDs&Vinyl, and Cell Phones&Accessories. For a fair comparison, we utilize the same strategy in the promising baseline DREM [9] to generate the training, valid and test data. The basic statistics of the datasets are shown in Table III.

2) Baselines: We compare our model to several typical baselines, including non-personalized (Group1), personalized (Group2) and explainable (Group3) search models. The details of the baselines are as follows:

(1) Non-personalized methods. The non-personalized search models do not take into account the user's personalized interest preferences when retrieving and returning relevant product items. We select the following typical non-personalized search models as our baselines. QL [28] uses a language model to rank documents based on the posterior probability of observing the query words. BM25 [29] ranks documents using a statistical scoring function that assumes a 2-Poisson distribution for the observed words. LaMART [30] ranks items using the ranking features extracted from the items' text and user behavior logs. LSE [11] ranks items by the similarity of item embeddings and their encoded n-grams from the reviews.

(2) Personalized methods: The personalized search models aim to retrieve and return personalized search results for users based on their individual preferences. The user's personalized interest preferences are usually captured from the user's search logs by various techniques, such as neural networks, attention mechanisms and etc. The following are typical personalized search models. AEM [3] constructs a query-aware attention mechanism to obtain dynamic user profiles for product search. ZAM [3] extends AEM with a zero attention mechanism for

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

TABLE IV
OVERALL PERFORMANCE OF MODELS

Method		Electronics			Kindle Store			CDs & Vinyl			Cell Phones		
		MAP	MRR	NDCG	MAP	MRR	NDCG	MAP	MRR	NDCG	MAP	MRR	NDCG
Group1	QL	0.289	0.289	0.316	0.011	0.012	0.013	0.009	0.011	0.010	0.081	0.081	0.092
	BM25	0.283	0.280	0.304	0.021	0.013	0.014	0.027	0.018	0.016	0.083	0.081	0.115
	LaMART	0.180	0.181	0.237	0.028	0.029	0.018	0.054	0.057	0.051	0.121	0.121	0.148
	LSE	0.233	0.234	0.239	0.006	0.007	0.007	0.018	0.022	0.020	0.098	0.098	0.084
Group2	AEM	0.265	0.265	0.290	0.024	0.025	0.028	0.032	0.038	0.037	0.105	0.106	0.145
	TEM	0.196	0.196	0.222	0.026	0.026	0.029	0.033	0.036	0.038	0.114	0.112	0.147
	ZAM	0.286	0.287	0.314	0.027	0.026	0.030	0.030	0.035	0.035	0.102	0.101	0.141
	HEM	0.308	0.309	0.329	0.029	0.035	0.033	0.034	0.040	0.040	0.124	0.124	0.153
	TransE	0.313	0.312	0.348	0.032	0.033	0.040	0.046	0.048	0.046	0.117	0.119	0.150
	SBG	0.398	0.402	0.437	0.065	0.071	0.070	0.076	0.081	0.093	0.261	0.264	0.323
	CAMI	0.402	0.403	0.461	0.081	0.080	0.069	0.079	0.082	0.098	0.284	0.283	0.351
Group3	PGPR	0.378	0.379	0.414	0.050	0.048	0.051	0.077	0.076	0.083	0.242	0.241	0.279
	DREM	0.366	0.367	0.408	0.057	0.067	0.067	0.074	0.084	0.086	0.249	0.249	0.282
	DREM-HGN	0.405	0.406	0.471	0.087	0.085	0.070	0.076	0.082	0.092	0.294	0.295	0.362
	QEPS	**0.420***	**0.422***	**0.525***	**0.112***	**0.116***	**0.074***	**0.081***	**0.091***	**0.176***	**0.331***	**0.333***	**0.408***

* Means our model outperforms all baselines with paired t-test at $p < 0.001$ level.

product search. TEM [2] replaces the zero attention in ZAM with a transformer [31] for product search. HEM [1] ranks items based on their posterior probability given the user and the product search query. Furthermore, our technical task is to predict the missing $item$ for the triplet $(user, query, ?)$ in the user-product KG, which is the link prediction task in the field of knowledge representation learning. The link prediction task aims at predicting the missing entity $h(t)$ or relation r for a triplet (h, r, t) in KGs. Thus, we select the typical knowledge representation learning method TransE [15], which achieves better performance in the link prediction task, as our personalized baselines.

(3) Explainable methods: The study of explainable retrieval systems has recently attracted the attention of researchers. Most of the existing studies on explainable IR focus on recommendation tasks, which focus on providing pre-hoc or post-hoc explanations for recommendation results. In the field of explainable recommendation, the reinforcement learning-based methods are relevant to our task. As a result, we select the state-of-the-art RL-based path reasoning method PGPR [6] for product recommendation as our baseline for comparison. PGPR is developed to explore the reasoning paths between users and items as explanations based on the user-conditional state in each step. For better use of PGPR in search scenarios, we also initialize the embeddings of the user-product KG by DREM [9], and further add the user's current query to the state of each step and make the next action selection based on the query enhanced user-conditional state. Moreover, in the product search domain, DREM [9] is a state-of-the-art explainable product search retrieval model. It ranks on the plausibility of the triplet $(user, query, item)$ in the user-product KG, and uses the entity soft matching to generate the path-formed explanations.

3) Implementation Details: In the training stage, we initialized the embeddings of the entities and relations of user-product KG by DREM [9] and set the maximum path length $T = 3$, the history length of h_t as 2, the pruned action space with the maximum size $\epsilon = 250$ and the maximum size of extracted path demonstrations $\alpha = 30$. For the hyperparameters, we select the learning rate $\mu \in \{0.0001, 0.001, 0.01, 0.1\}$, the embedding dimension $d \in \{100, 200, 400\}$, the batch size $B \in \{64, 512, 1024\}$, the weights in the reward functions $\{\lambda_0, \lambda_1, \lambda_2, \lambda_3\} \in [0, 1]$ with interval 0.1. The optimal hyperparameter configuration is determined by grid search as follows: the learning rate $\mu = 0.0001$, the embedding dimension $\{d, d_a\} = 400$, the batch size $B = 512$, the reward weights $\lambda_1 = 0.98$, $\lambda_4 = 0.01$ for CellPhones, $\lambda_1 = 0.97$, $\lambda_4 = 0.02$ for Electronics and CD, $\lambda_1 = 0.96$, $\lambda_4 = 0.03$ for Kindle and $\lambda_2 = \lambda_3 = 0.005$ for all datasets. In the testing stage, we only retrieve 100 items to generate the ranked list for each user-query pair. Following the baselines in our paper, we compute Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain at 10 (NDCG@10) to evaluate the retrieval performance in the experiments.

B. Retrieval Result and Analysis

Table IV summarizes the overall performance on the Amazon dataset, showing that our model consistently outperforms all baselines on all metrics across four datasets. It confirms that our model successfully leverages the query-aware rewards and the demonstration-guided policy network to encourage the agent to retrieve the appropriate products and persuasive explanations. The key to the improvement in retrieval performance is that the search behaviors related to the current query are well explored in the path-finding process. Specifically, for the Electronics/ Kindle Store/ CD&Vinyl/ Mobile Phones datasets, our model achieves at least 0.015/ 0.025/ 0.002/ 0.037 higher performance on MAP, 0.016/ 0.031/ 0.004/ 0.038 higher performance on MRR and 0.054/ 0.004/ 0.078/ 0.046 higher performance on NDCG than state-of-the-art baseline methods. Furthermore, we can see that: (1) Personalized models achieve higher retrieval performance than the non-personalized models. This is because the personalized models take full advantage of the user's historical behavior logs to capture the dynamic and personalized user profiles for more accurate retrieval results. While, TransE has comparable or better results than other personalized models because it utilizes additional meta-information (i.e., item attribution and item-item interactions) in the KGs to build users' search intentions. (2)

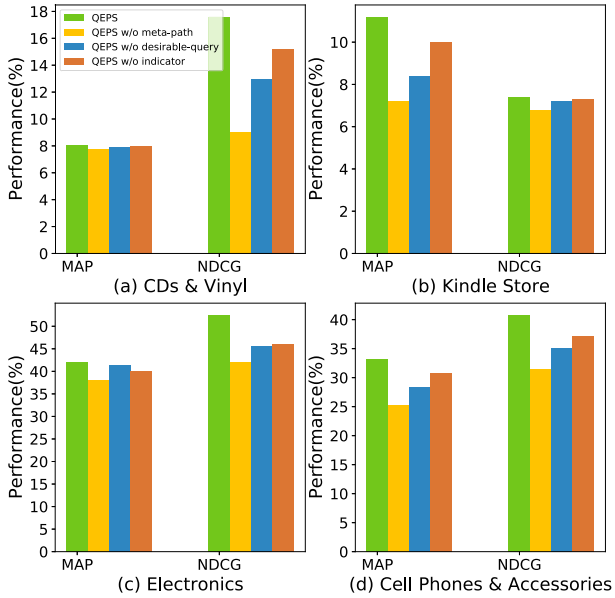

Fig. 3. Performance of different variants of QEPS.

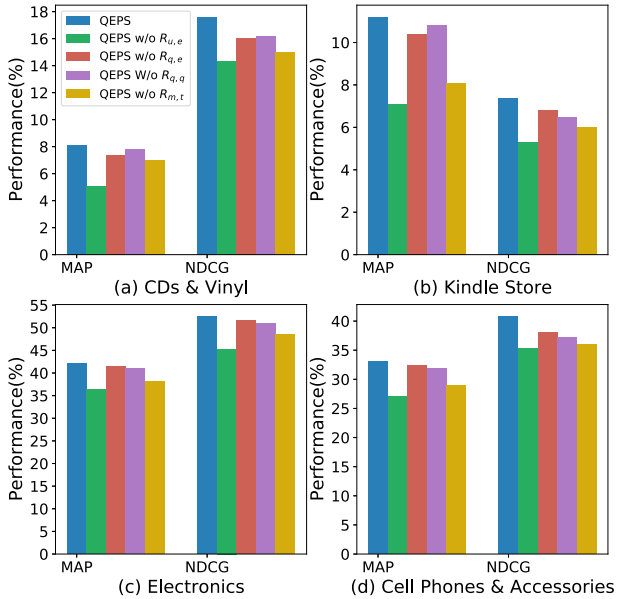

Fig. 4. Performance of different rewards of QEPS.

Compared with the personalized and non-personalized models, the explainable models obtain the best performance. It indicates that the path reasoning in the explainable models can effectively integrate the auxiliary information from the user-product KG to enhance performance. (3) Among the explainable models, our model QEPS outperforms PGPR and DREM. This suggests that our model implements an efficient search strategy in path inference, while PGPR and DREM may suffer from noise in their reasoning paths. The main reason for this is that QEPS uses query-aware rewards and demonstration-aware guidance for path-finding process, which can explore reasonable paths to correct items using the search behaviors associated with the current query. (4) Compared with the graph-based baselines such as DREM and SBG, our model has a better retrieval performance on four datasets. This is due to the fact that we consider the demonstration guidance on the path exploration for reaching the correct products.

Ablation study: Compared to the baselines, our model QEPS develops a demonstration-guided policy network featured by query-specific path demonstrations and query-aware rewards. The path demonstrations are extracted by the demonstration extractor based on the manual meta-paths, aiming to provide supervised signals with the demonstration-based indicator on the path-finding process. To explore the demonstration-based components on the effectiveness of our model, i.e., demonstration extractor and demonstration-based indicator, we construct several variants as: (1) QEPS w/o meta-path does not consider the metapaths defined in Table II, and generates the path demonstrations with random walks. (2) QEPS w/o desirable-query uses the defined meta-paths to generate the demonstrations without considering the relevance of the demonstrations to the current query, i.e., (5). (3) QEPS w/o indicator means that QEPS does not use the demonstration-based indicator to guide the path-finding process of the actor-critic policy network.

The experimental results are shown in Fig. 3, which are: (1) QEPS w/o meta-path has a lower retrieval performance than QEPS. This indicates that our manually defined meta-paths can facilitate the model's search for the correct target products compared to the randomly sampled paths. (2) QEPS w/o desirable-query achieves worse retrieval performance than QEPS, suggesting that the relevance (similarity) modeling between the *desirable-query* in the demonstrations and the current query is necessary to find stronger path-based explanations.

(3) Compared with QEPS w/o indicator, QEPS obtains better retrieval performance. This means that the demonstration-based indicator can provide valuable guidance in the path-finding process to arrive at correct products with convincing explanations.

Moreover, to analyze the importance of different rewards in QEPS, we construct QEPS w/o $R_{u,e}$, QEPS w/o $R_{q,e}$, QEPS w/o $R_{q,q'}$ and QEPS w/o $R_{m,t}$, which does not consider the knowledge-based reward $R_{u,e}$, the semantic-based reward $R_{q,e}$, the path-based reward $R_{q,q'}$ and the immediate reward $R_{m,t}$, respectively. Fig. 4 gives the convincing experimental results, which are: (1) QEPS achieves the highest retrieval performance among all variants. It is suggested that the four rewards are all necessary to guide the agent to find the correct product items for the users. (2) QEPS w/o $R_{u,e}$ achieves the lowest retrieval results among the variants. It indicates that the knowledge-based reward is the most important signal to explore the correct target products. (3) QEPS w/o $R_{q,e}$ and w/o $R_{q,q'}$ has slightly lower performance than QEPS. It indicates that the rationalization of the reasoning path and the semantic information of the items' context is the valuable signal to guarantee the high-quality of retrieved results and explanations. (4) QEPS w/o $R_{m,t}$ achieves lower performance than QEPS, suggesting that the immediate reward from the demonstration-based indicator is effective in finding correct paths to the items that can satisfy the user's current search intent.

User and query	Path-based explanation
Case1: user *APX* with the submitted query *phone screen protector*	User: APX —q_1:{phone accessory}→ Home Button Sticker for iPhone —q_1:{phone accessory}→ User: A10 —q_2:{phone screen protector }→ Screen Protector for iPhone 4
Case 2: user *AN0* with the submitted query *thriller and suspense ebook.*	User: AN0 —mention→ Fiction —mention$^-$→ User: A11 —q_3:{mystery and sleuth fiction }→ Coffin tales season of death
Case 3: user *ANS* with submitted query *accessory headphone*	User: ANS —q_4:{electronics earphones}→ DJ Headphones —producedby→ HiFiMan —producedby$^-$→ Stereo Headphone
Case 4:user *A3I* with the submitted query *classic rock music*	User: A3I —q_5:{cds classic rock}→ Brain salad surgery(CD name) —describedAs→ rock —describedAs$^-$→ Lead Guitar with Wolf Marshall
Case 5: user *A28* with the submitted query *romance historical ebooks*	User: A28 —q_6:{mystery sleuth ebook}→ The Girl Before —alsoViewed→ The Night Fire —alsoViewed$^-$→ The Mysterious MR Quin
Case 6: user *A3B* with the submitted query *electronics navigation GPS accessory*	User: A3B —q_7:{automotive accessories}→ Battery pack change equipment —belongsTo→ ZOTAC —belongsTo$^-$→ Rugged GPS handheld
Case 7: user *AKP* with the submitted query *vinyl children music*	User: AKP —q_5:{upbeat animal music}→ Five little momkeys —boughtTogether→ Row row your boat —boughtTogether$^-$→ Did you ever see my tail
Case 8: user *ASZ* with the submitted query *cds vinyl metal*	User: ASZ —q_5:{cds vinyl metal}→ Take Me Back To Eden

Fig. 5. Real cases of reasoning paths for product search.

C. Usefulness of Relevant Search Behaviours

In our model, the user's relevant search behaviors are those *desirable-query* with high similarity to the user's current query in the reasoning path, which are the key to boosting explanations. For example, given user A and the current query *maybelline remover*, the reasoning path *user A* $\xrightarrow{mention}$ *fashion&gentle* $\xrightarrow{mention^-}$ *user B* $\xrightarrow{q:\{maybelline\ remover\}}$ *maybelline EYE+LIP makeup remover* can provide the meaningful explanations because the *desirable-query* builds a strong correlation with the current query based on their high-similarity. The explanation can be as *the product is retrieved because user B with a similar taste to the target user A previously purchased this product by the current query*.

To explore the importance of the relevant search behaviors on the retrieval performance, we heuristically select triplets $\{(u', q', i')\}$ with their scores greater than a threshold δ as the relevant search behaviors for the current queries $\{q\}$ in the train set. The scores are the cosine similarity of the queries q' and q based on their semantic representations calculated as equation (1). In the experiment, we empirically set $\delta = 0.95$ and generate 3,086/12,031 triplets for the smallest and largest datasets, i.e., Cell Phones and Electronics.

We then use these triplets with the rate $\alpha = \{0\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$ in the training stage. Fig. 6 gives convincing results, which are: (1) when $\alpha = 0\%$, the relevant search queries are not used in the reasoning process. QEPS has the lowest accuracy because it does not extract valuable information from the relevant search behavior to find paths to the correct items. (2) With the growth of α, the performance of QEPS consistently increases and outperforms DREM, i.e. the richer the relevant search behaviors, the higher the accuracy. This confirms that the relevant search behaviors in KGs do indeed help to improve retrieval performance and further increase explanations.

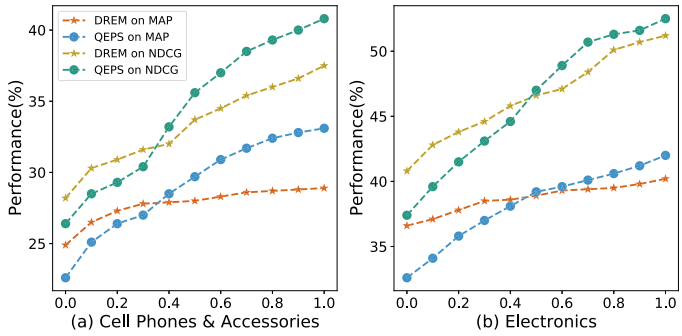

Fig. 6. Performance on the relevant search behaviors.

Case study. To intuitively understand how the relevant search behaviors affect the product search, we give a case study in Fig. 5, where the queries with red font in the paths are the search behaviors related to the user's current queries. Specifically, (1) in case 1, two users have similar search behaviors because they both previously purchased the product *Home Button Sticker for iPhone* under the query *phone accessory*. When the target user submits *phone screen protector*, QEPS returns the product *Screen Protector for iPhone 4* with the explanation as *the product is retrieved because the user with similar search behavior to the target user previously purchased this product under the submitted query*.

(2) Both users in case 2 have similar tastes as they like something in fiction. Based on the high-similarity of the query $q_3 : \{mastery\ and\ sleuth\ fiction\}$ and the submitted query *thriller and suspense ebook*, the product *Coffin tales season of death* is returned with the explanation as *the product is retrieved because the user with similar tastes to the target user previously purchased this product under the similar query*. (3) Conditional on the high similarity between the query q_4 :

TABLE V
CROWDSOURCING RESULTS FOR EXPLANATION EVALUATION

Model	Informativeness	Satisfaction	Usefulness
Equal	10%	16%	12%
DREM-HGN	28%	30%	43%
QEPS	62%	54%	45%

$\{electronics\ Earphones\}$ and the submitted query *accessory headphone*, in case 3 the explanation is returned as *the product is retrieved because the user has previously purchased products from brands such as HiFiMan under the similar query*. In general, these typical explanations of cases illustrate that the search behaviors related to the current query in the reasoning paths can provide a bridge to generate a reasonable explanation of the search results.

D. Explanation Evaluation

Our evaluation of the search explanations focuses on three main perspectives: (1) Informativeness, assessing whether the explanations provided relevant information about the item and the query; (2) Usefulness, determining if the explanations were effective in attracting users to purchase the item; and (3) Satisfaction, gauging whether providing explanations increased users' satisfaction with the product search engine's service. In this paper, we conduct pairwise comparisons between explanations generated by QEPS and the promising explainable method DREM-HGN for each user-query-item combination. We ask workers to annotate their preferences based solely on pairwise comparisons. Pairwise preferences have been proven to be much more robust and reliable compared to pointwise relevance judgments in IR [32]. In this way, we conduct a user study to enhance the quality of our crowdsourcing experiments. First, our crowdsourcing dataset is derived from the Electronics retrieval experiment dataset. Electronics is a popular product category on Amazon, known for having less complex knowledge structures. Inspired by DREM-HGN [33], we randomly select 101 user-query pairs from the Electronics test data, where both QEPS and DREM-HGN achieved MRR scores greater than or equal to 0.1.

To ensure fairness, we create user-query-item triples by pairing user-query pairs with the item that was actually purchased by the user in the corresponding search session. Consequently, all sampled items are actually purchased by the user, and the workers are tasked with judging which explanations better explain the user's purchase behavior in the search session. To obtain reliable labels, we recruit three workers per case and use a voting process to determine the final annotations. Moreover, to ensure that the evaluation process was impartial and unbiased, we employ a strategy to anonymize QEPS and DREM-HGN by randomly labeling them as "Explanation A" and "Explanation B". The workers' task is simply to determine which explanation presents better search explanations: "Explanation A", "Explanation B", both, or none. Table V presents the results of our crowdsourcing experiment, where most workers preferred the explanations provided by QEPS over DREM-HGN in terms of Informativeness and Satisfaction. QEPS's ability to capture the user's search intentions through the retrieval model's inference process contributes to its reliability and trustworthiness. However, there were no significant differences between QEPS and DREM-HGN in terms of Usefulness, although QEPS had slightly higher overall scores in this aspect.

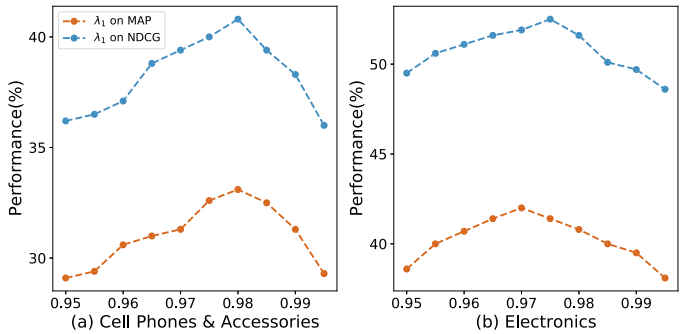

Fig. 7. Performance with different weights of knowledge-based reward.

E. Discussion

1) Reward Weight Sensitivity: To analyze the sensitivity of the reward weights, we plot MAP, NDCG metrics of QEPS on Cell Phones and Electronics datasets with different parameter settings. Based on the optimal settings in subsection IV-A3, we fix $\lambda_2 = \lambda_3 = 0.005$ and report the results for the reward weight $\lambda_1 \in [0.95, 1.0]$ with interval 0.005. Fig. 7 gives the results: (1) When $\lambda_1 = 0.99$, QEPS is degraded to QEPS w/o $R_{m,t}$, which achieves slightly worse performance than QEPS. (2) QEPS achieves worse performance with smaller or larger λ_1. It is because that too small λ_1 may be difficult to provide valuable information for the agent to explore reasoning paths, and too large λ_1 may introduce much more noise than the useful signals. (3) QEPS achieves the best performance when $\lambda_1 = \{0.98, 0.97\}$ on two datasets. It reveals that both query-aware knowledge-based terminal reward and demonstration-guided immediate reward are both valuable in encouraging the agent to reach the correct product items.

2) Effectiveness of Pruning Strategy: How to determine the concise and precise action space for different states is the challenge in the RL-based algorithm. In our model, it is more important to simplify the action space for the entities with lots of neighbors. Thus we design the following pruning strategies to evaluate the effectiveness of our pruning function: (1) Random pruning: we randomly select actions from the entitie's neighbors, (2) Simplified pruning: we directly use $f((r', e')|u) = (\mathbf{u} + \mathbf{r}') \odot \mathbf{e}'$ to make the action selection. The experimental results in Fig. 8, where we can see that: (1) our model with the pruning action function in Eq (2) achieves the highest retrieval performance among the pruning variants, suggesting that the pruning function considers the plausibility of the reasoning path during the path-finding process is helpful to obtain remarkable retrieval performance. (2) The simplified pruning strategy has better performance than the random one. This is because random

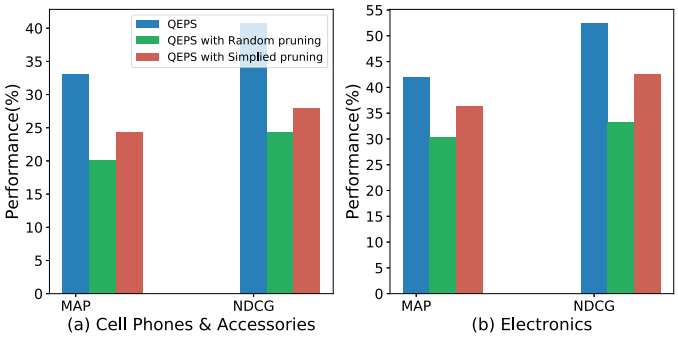

Fig. 8. Performance with different pruning functions.

TABLE VI
COMPUTATIONAL COMPLEXITY

Model	Computational complexity
DREM	$O((N_e + N_r) * d)$
DREM-HGN	$O((N_e + N_r + N_T) * d + N_T * d^2)$
PGPR	$O(d * d_s + d_s^2 + N_a * d_a)$
QEPS	$O(d * d_s + d_s^2 + N_a * d_a)$

pruning may filter out the valuable actions that contribute to reaching the correct products.

3) Computational Complexity: In our model, the learned embeddings are the parameters of the actor, critic and demonstration-based indicator. Thus the time complexity of our model and the baselines RGPR, DREM and DREM-HGN are $O(d * d_s + d_s^2 + N_a * d_a)$, $O(d * d_s + d_s^2 + N_a * d_a)$, $O((N_e + N_r) * d)$ and $O((N_e + N_r + N_T) * d + N_T * d^2)$. We make the comparison of computational complexity between our model and the baselines in Table VI. In this table, d, d_s, d_a are the embedding dimension of the entity (relation), state and action in the policy network, respectively. N_a, N_e, N_r, N_T are the number of the actions, entities, relations and entity types. After analyzing, it usually exists $N_e + N_r \gg N_a \gg d_s \geq d = d_a > N_T$, where we can see that our model has a similar computational complexity with state-of-the-art RL-based baselines, and the lower computational complexity with increasing $(N_e + N_r)$ and N_T.

V. RELATED WORK

The field of explainable artificial intelligence is young and has recently received considerable attention in industry and academia. Focusing on the concerns of this paper, there are two lines of studies that are related to our work: explainable recommendation and explainable search.

A. Explainable Recommendation

The explainable recommendation aims to explore why the items are recommended, i.e., how a recommended item relates to a user's preferences [34]. Providing explanations has been shown to have great advantages in improving algorithm transparency and user satisfaction [35]. In the field of explainable recommendation, there are many methods that can be roughly classified into user-review explanations [36], [37], [38], image-visualizations [39], reasoning rules [40], [41] and knowledge-aware explanations [42], [43], [44], etc. Among these methods, the user-review explanations [36], [37] usually highlighted the words or sentences with some strategies (e.g., attention mechanism and pre-defined templates) in the user review information as explanations. Different from the user-review explanations, the knowledge-aware explanations [6], [7] mainly performed path reasoning [45], [46], [47], [48] over the knowledge graph, where the reasoning paths between user and item were constructed to generate path-formed explanations. More specifically, the reasoning process can be explored in various algorithms, such as recurrent neural network [4], [5], graph convolutional neural network [7], [49], [50], [51], and reinforcement learning [52], etc., for which the RL-based reasoning methods are more promising and effective. In summary, the RL-based reasoning methods formulate multi-hop reasoning as a sequential decision making problem. For example, the promising RL-based reasoning model PGPR [6] conducted path reasoning by reinforcement learning (RL) technique, which is featured by the user-conditional state and rewards. Due to these RL-based knowledge reasoning methods on explainable recommendation are closely related to our work, we select PGPR [6] as our baseline for comparison. Importantly, a fundamental difference between this work and our model QEPS is that we leverage the query-aware rewards and demonstrations to find paths to correct items and explore the relevant search behaviors to boost explanations.

B. Explainable Search

In the search scenario, the user's intents can be explicitly expressed through queries, which is fundamentally different from recommendations. With the focus on the explainable search field, the existing methods mainly focus on retrieving text documents based on the user's query, such as news articles or web pages. For example, SHAP [53] explores the importance of input features on the model's prediction results to explain the output of the model. Unlike document-based search, which focuses on text matches, product search is more expert in using information such as knowledge entities and user history logs to determine user purchases. In general, product search aims to retrieve and return relevant products to customers based on their submitted queries.

Previous studies [54], [55], [56], [57], [58], [59] mainly use products' aspects (e.g., brand, category, context) to do non-personalized search for users. The typical DP [54] retrieved products by matching queries with multiple aspects of products simultaneously. LSE [11] retrieved products by matching queries and products with their latent representations. With the increasing complexity of user needs, many personalized models [1], [3], [60], [61], [62] have emerged to capture users' personal interests and return user-centric products. The popular personalized models like [1], [2], [3] designed a variety of attention mechanisms (e.g., zero-attention, self-attention) to aggregate users' historical behaviors with queries as user's profiles and use them to retrieve personalized products.

Recently, explainability is an important criterion to measure the quality of a product system, which has been extensively studied for product recommendation. Although numerous methods using KGs for explainable recommendations have been successful, few studies have been done to explain search results in the product search scenario. DREM [9] made the first attempt to exploit the structural information in user-product KGs for explainable product search. It built the queries as dynamic relations between users and products and adopted entity soft matching with knowledge embeddings to extract the post-hoc soft explanation. The fundamental difference between DREM and our model is that (1) our search results are produced by the reasoning process, while DREM is not, and (2) our explanations are tailored to the current search intentions of users, and are more explicit and persuasive than DREM.

VI. Conclusion

This paper proposes a Query-aware Explainable Product Search with Reinforcement Knowledge Reasoning QEPS for explainable product search. QEPS develops a demonstration-guided policy network, which is characterized by the query-aware rewards and path demonstrations to return correct products to users as well as the reasoning path to explain the retrieved products. The query modeling in the rewards and demonstration guidance on the policy network can well explore the user's relevant search behavior with the aim of providing more accurate products and convincing explanations to the users. Empirical experiments on four datasets show that our model achieves remarkable performance and has the ability to generate reasonable explanations for search results.

Acknowledgment

The work was partially done at the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE, and Beijing Key Laboratory of Big Data Management and Analysis Methods.

References

[1] Q. Ai, Y. Zhang, K. Bi, X. Chen, and W. B. Croft, "Learning a hierarchical embedding model for personalized product search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 645–654.

[2] K. Bi, Q. Ai, and W. B. Croft, "A transformer-based embedding model for personalized product search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1521–1524.

[3] Q. Ai, D. N. Hill, S. V. N. Vishwanathan, and W. B. Croft, "A zero attention model for personalized product search," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 379–388.

[4] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, AAAI Press, 2019, pp. 5329–5336.

[5] Q. Zhu, X. Zhou, J. Wu, J. Tan, and L. Guo, "A knowledge-aware attentional reasoning network for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, AAAI Press, 2020, pp. 6999–7006.

[6] Y. Xian, Z. Fu, S. Muthukrishnan, G. de Melo, and Y. Zhang, "Reinforcement knowledge graph reasoning for explainable recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 285–294.

[7] Y. Xian et al., "CAFE: Coarse-to-fine neural symbolic reasoning for explainable recommendation," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1645–1654.

[8] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, "Learning heterogeneous knowledge base embeddings for explainable recommendation," *Algorithms*, vol. 11, no. 9, 2018, Art. no. 137.

[9] Q. Ai, Y. Zhang, K. Bi, and W. B. Croft, "Explainable product search with a dynamic relation embedding model," *ACM Trans. Inf. Syst.*, vol. 38, no. 1, pp. 4:1–4:29, 2020.

[10] K. Zhao et al., "Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 239–248.

[11] C. V. Gysel, M. de Rijke, and E. Kanoulas, "Learning latent vector spaces for product search," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 165–174.

[12] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for top-n recommendation with a neural co-attention model," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1531–1540.

[13] S. Fan et al., "Metapath-guided heterogeneous graph neural network for intent recommendation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2478–2486.

[14] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003.

[15] A. Bordes, N. Usunier, and A. Garcia-Dur 'an, "Translating embeddings for modeling multi-relational data," in In *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.

[16] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI Conf. Artif. Intell.*, AAAI Press, 2014, pp. 1112–1119.

[17] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. Assoc. Comput. Linguistics*, 2015, pp. 687–696.

[18] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 2071–2080.

[19] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," 2014, *arXiv:1412.6575*.

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[21] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 55–64.

[22] S. Ge, Z. Dou, Z. Jiang, J. Nie, and J. Wen, "Personalizing search results using hierarchical RNN with query-aware attention," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 347–356.

[23] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Representations*, 2013.

[25] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2016.

[26] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, pp. 9–44, 1988.

[27] J. Yao, Z. Dou, J. Xu, and J. Wen, "RLPS: A reinforcement learning-based framework for personalized search," *ACM Trans. Inf. Syst.*, vol. 39, no. 3, pp. 27:1–27:29, 2021.

[28] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," *SIGIR Forum*, vol. 51, no. 2, pp. 202–208, 2017.

[29] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1994, pp. 232–241.

[30] C. Wu, M. Yan, and L. Si, "Ensemble methods for personalized e-commerce search challenge at CIKM cup 2016," 2017, *arXiv: 1708.04479*.

[31] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[32] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *Proc. 28th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2017, pp. 4–11.

[33] Q. Ai and L. R. Narayanan, "Model-agnostic vs. model-intrinsic interpretability for explainable product search," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 5–15.
[34] K. Balog, F. Radlinski, and S. Arakelyan, "Transparent, scrutable and explainable user models for personalized recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 265–274.
[35] X. Chen, Y. Zhang, and J. Wen, "Measuring "why" in recommender systems: A comprehensive survey on the evaluation of explainable recommendation," 2022, *arXiv:2202.06466*.
[36] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2017, pp. 425–434.
[37] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proc. Int. Conf. World Wide Web*, 2018, pp. 1583–1592.
[38] H. Chen, X. Chen, S. Shi, and Y. Zhang, "Generate natural language explanations for recommendation," 2021, *arXiv:2101.03392*. [Online]. Available: https://arxiv.org/abs/2101.03392
[39] X. Chen et al., "Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 765–774.
[40] Y. Zhu, Y. Xian, Z. Fu, G. de Melo, and Y. Zhang, "Faithfully explainable recommendation via neural logic reasoning," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2021, pp. 3083–3090.
[41] S. Shi, H. Chen, W. Ma, J. Mao, M. Zhang, and Y. Zhang, "Neural logic reasoning," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1365–1374.
[42] D. Liu, J. Lian, Z. Liu, X. Wang, G. Sun, and X. Xie, "Reinforced anchor knowledge graph generation for news recommendation reasoning," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1055–1065.
[43] G. Balloccu, L. Boratto, G. Fenu, and M. Marras, "Post processing recommender systems with knowledge graphs for recency, popularity, and diversity of explanations," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 646–656.
[44] Y. Yang, J. Lin, X. Zhang, and M. Wang, "PKG: A personal knowledge graph for recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 3334–3338.
[45] A. García-Durán, A. Bordes, and N. Usunier, "Composing relationships with translations," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2015, pp. 286–290.
[46] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2015, pp. 705–714.
[47] Q. Zhu, X. Zhou, J. Tan, and L. Guo, "Knowledge base reasoning with convolutional-based recurrent neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2015–2028, May 2021.
[48] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 635–644.
[49] Z. Zhang, Z. Li, H. Liu, and N. N. Xiong, "Multi-scale dynamic convolutional network for knowledge graph embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2335–2347, May 2022.
[50] Z. Li, Q. Zhang, F. Zhu, D. Li, C. Zheng, and Y. Zhang, "Knowledge graph representation learning with simplifying hierarchical feature propagation," *Inf. Process. Manage.*, vol. 60, no. 4, 2023, Art. no. 103348.
[51] Z. Li, Y. Zhao, Y. Zhang, and Z. Zhang, "Multi-relational graph attention networks for knowledge graph completion," *Knowl.-Based Syst.*, vol. 251, 2022, Art. no. 109262.
[52] S. Zhou et al., "Interactive recommender system via knowledge graph-enhanced reinforcement learning," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 179–188.
[53] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4765–4774.
[54] S. C. J. Lim, Y. Liu, and W. B. Lee, "Multi-facet product information search and retrieval using semantically annotated product family ontology," *Inf. Process. Manage.*, vol. 46, no. 4, pp. 479–493, 2010.
[55] P. Nurmi, E. Lagerspetz, W. L. Buntine, P. Floréen, and J. Kukkonen, "Product Retrieval for Grocery Stores," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008, pp. 781–782.
[56] Y. Guo, Z. Cheng, L. Nie, X. Xu, and M. S. Kankanhalli, "Multi-modal preference modeling for product search," in *Proc. Multimedia Conf.*, 2018, pp. 1865–1873.
[57] K. Bi, C. H. Teo, Y. Dattatreya, V. Mohan, and W. B. Croft, "Leverage implicit feedback for context-aware product search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019.
[58] S. K. K. Santu, P. Sondhi, and C. Zhai, "On application of learning to rank for e-commerce search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 475–484.
[59] X. He, T. Chen, M. Kan, and X. Chen, "TriRank: Review-aware explainable recommendation by modeling aspects," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 1661–1670.
[60] Y. Guo, Z. Cheng, L. Nie, Y. Wang, J. Ma, and M. S. Kankanhalli, "Attentive long short-term preference modeling for personalized product search," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 19:1–19: 27, 2019.
[61] K. Bi, Q. Ai, Y. Zhang, and W. B. Croft, "Conversational product search based on negative feedback," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 359–368.
[62] L. Wu, D. Hu, L. Hong, and H. Liu, "Turning clicks into purchases: Revenue optimization for product search in e-commerce," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 365–374.

Qiannan Zhu received the PhD degree from the Institute of Information Engineering, Chinese Academy of Sciences, in 2020, and the postdoctoral degree from the Gaoling School of Artificial Intelligence, Renmin University of China, in 2023. Currently she is a lecturer with the School of Artificial Intelligence, Beijing Normal University. Her research interests include recommendation system, information retrieval, knowledge representation and large language models.

Haobo Zhang is working toward the PhD degree with the Gaoling School of Artificial Intelligence, Renmin University of China. His research interests include explainable recommendation, product search and information retrieval.

Qing He is working toward the degree with the School of Finance, Renmin University of China. Her research interests include explainable recommendations, product search, and information retrieval.

Zhicheng Dou (Member, IEEE) received the BS and PhD degrees in computer science and technology from Nankai University, in 2003 and 2008, respectively. He is a professor with the Gaoling School of Artificial Intelligence, Renmin University of China. He worked with Microsoft Research as a researcher from 2008 to 2014. His research interests include information retrieval, web search, and nature language processing.