Counterfactual Path Augmentation for Reinforcement Reasoning in Explainable Recommendation

Yue Kou[1], Eryu Jiang[1], Derong Shen[1], Xiangmin Zhou[2], Dong Li[3](✉), Tiezheng Nie[1], and Ge Yu[1]

[1] Northeastern University, Shenyang, Liaoning 110004, China
{kouyue,shenderong,nietiezheng,yuge}@cse.neu.edu.cn,2201804@stu.neu.edu.cn
[2] RMIT University, Melbourne, VIC 3000, Australia
xiangmin.zhou@rmit.edu.au
[3] Liaoning University, Shenyang, Liaoning 110036, China
dongli@lnu.edu.cn

Abstract. Most graph-based explainable recommender systems employ paths within knowledge graphs to provide explanations for the recommended items. However, existing technologies often fail to provide detailed explanations for these paths, making it difficult to elaborate in a fine-grained manner on why a particular path is selected, specifically which relations along the path have played a crucial role. In this paper, we propose an explainable recommendation model with counterfactual path augmentation for reinforcement reasoning. Specifically, we propose a user preference learning method based on counterfactual path augmentation, which leverages counterfactual reasoning to learn the degree of trustworthiness that users assign to various candidate paths and even to the individual relations within the paths. Then we propose a dual-reward reinforcement learning approach for generating recommendations and explanations. This method integrates path-oriented rewards with item-oriented rewards to simultaneously enhance the accuracy and explainability of the model. Finally, we propose two novel evaluation metrics, namely stability and effectiveness, to evaluate explainability quality. We evaluate our model on four real-world datasets and the experimental results show the superiority of our model compared to state-of-the-art recommendation models.

Keywords: Explainable recommendation · Counterfactual path augmentation · Reinforcement reasoning · Knowledge graph.

1 Introduction

The explainability of recommender systems is crucial, as in recommendation scenarios, it is difficult to conclusively assert that one recommendation is flawless

This work was supported by the National Natural Science Foundation of China (62072084, 62472204 and 62172082) and the Fundamental Research Funds for the Central Universities under Grant No.N2116008.

while others are entirely incorrect. In such cases, clear and reasonable explanations become crucial, as they help users understand the rationale behind the recommendation of a particular item, thus improving the precision, transparency, and trustworthiness of the recommendations, and facilitating users to make more informed decisions [4,6]. Among them, the explainable approach using graphical structured displays is currently a highly regarded strategy for explainable recommendation. Compared with traditional recommendation methods, such as those based on collaborative filtering, the abundant entity and structural information contained within knowledge graphs provides a more solid foundation for models to generate more reliable explanations. The core of this method lies in elucidating the rationale behind the recommendations results by deducing a path that starts from the target user and ends at the target item, as it believes that these paths can preserve causal relationships in the real world [13, 15, 18]. However, the complexity of the graph results in numerous potential paths influencing the final decision, making it vital to enrich path information and refine the path generation process.

We study the problem of graph-based explainable recommendations. Given a target user and the corresponding knowledge graph, our goal is to leverage path optimization techniques and reinforcement reasoning mechanisms to accurately recommend items that meet user demands while simultaneously providing paths as explanations that can meticulously exhibit user preference. For explainable graph-based recommendations, three key issues need to be addressed. (1) When there are multiple paths in the knowledge graph that can serve as explanations (i.e., paths from the target user to the target item), we need to select the path that is most convincing to the user and provide reasons for such selection. As shown in Fig.1, $path_1$ and $path_2$ are two candidate paths which might serve as explanations for recommending a foundation product to the user, conveying the meanings of "it is from the same brand as the lipstick you have purchased" and "it belongs to the same foundation category as the cushion foundation you have purchased," respectively. Assuming that the user's preference is to focus on brand information rather than category information, then $path_1$ should be chosen as the explanation because it contains brand relations and aligns more closely with the user's preference. Therefore, a good model should not only be able to select explanation paths that better match user preferences but also explain in detail why a specific path is chosen, including which relations within the path have played pivotal roles. (2) We need to design a novel model for generating the recommendation that not only effectively reflects user preferences but also provides convincing explanations to users. While the preferences of users infer the probability that a recommendation is accepted by them, the credibility of the explanations determines whether users will trust the recommendation results. As shown in Fig.1, based on the user's purchase of air cushion foundation, we derive $path_2$ for liquid foundation (same category) and $path_3$ for makeup cotton (related to makeup removers). If the user thinks $path_3$ more credible, he/she may prefer makeup cotton, even if it matches his/her preferences slightly less well. Therefore, a good recommendation model should consider both the path credibility and user preferences to predict the products attracting him/her. (3) We need to effectively evaluate the quality of explainability. There should be a unified quantitative method for measuring the explainability of path-based models, as assessing whether an explanation is trustworthy is highly subjective. Most methods rely on case studies to evaluate explainability. To our knowledge, there are few widely accepted standards for assessing explainability. Therefore, enhancing the evaluation of explainability for path-based models is crucial.

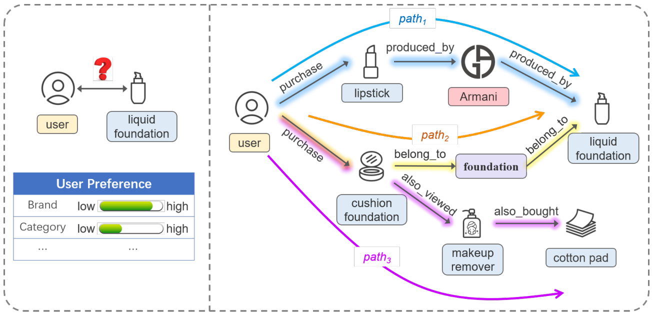

Fig. 1: Motivation Examples.

Previous explainable recommendation approaches can be categorized into reinforcement learning-based approaches [9,12,13,18] and counterfactual reasoning-based approaches [5, 7, 10, 11, 14]. However, none of them support the explainability of paths from a fine-grained perspective, nor do they consider the balance between path trustworthiness and recommendation accuracy. To overcome the problems of existing explainable recommendation approaches, we propose an explainable recommendation model with counterfactual path augmentation for reinforcement reasoning. Specifically, we can select the path that aligns best with user preferences as the explanation. Then both the credibility of explanations and the accuracy of recommendations will be considered together to predict the items' ratings. Finally, the quality of explainability can be evaluated more effectively. We summarize our contributions as follows.

- We propose a user preference learning method based on counterfactual path augmentation, which leverages counterfactual reasoning to learn the degree of trustworthiness that users assign to various candidate paths and even to the individual relations within the paths.
- We propose a dual-reward reinforcement learning approach for generating recommendations and explanations. By combining path-based rewards with item-focused rewards, this approach aims to simultaneously boost the model's accuracy and its ability to provide credible explanations.
- We propose two novel evaluation metrics, stability and effectiveness, to evaluate explainability quality. Compared to traditional metrics, our proposed met-

rics can reduce the interference of user subjective factors on the assessment results.

- We conduct extensive experiments on four real-world datasets and the experimental results demonstrate the high effectiveness of our proposed model for path-based explainable recommendation.

2 Related Work

We review existing work on three topics related to our work, including reinforcement learning-based explainable recommendation, Large Language Models-based explainable recommendation and counterfactual reasoning-based explainable recommendation.

Reinforcement Learning-based Explainable Recommendation. Many researchers have adopted reinforcement learning-based models for explainable recommendation, often explaining their results by tracing a path to the target item within a graph. For instance, [13] proposed a strategy-guided path reasoning approach to explicitly reason about paths in the decision-making process of knowledge graphs. To further enhance the effectiveness of path reasoning models, [18] developed a model that utilizes example paths to guide and supervise the path finding process. However, a limitation of this approach is that path selection heavily relies on expert knowledge. In contrast, [9] presented a user-centric path reasoning network that dynamically captures users' demand information, thereby enabling explainable recommendations. Additionally, [12] proposed a multi-level view-based reinforcement learning framework capable of modeling users' interests at various levels. While these methods effectively model user preferences and improve recommendation accuracy, they fail to provide detailed explanations for these paths.

Counterfactual Reasoning-based Explainable Recommendation. Counterfactual reasoning has also emerged as an explainable method in recent years [11]. It addresses the "hypothetical" causality question, which states that if a certain condition had not occurred, the outcome would have been different. For example, [10] proposed a model-agnostic explainable proposal for counterfactual reasoning by learning slight perturbations. [7] proposed a similar model-agnostic explainable method, perturbing user features to flip model decisions to learn counterfactual explanations. To leverage the rich recommendation data, [14] used the review data of items to enrich the information for training counterfactual inference. [5] proposed a novel explainable framework for path-based recommendation using counterfactual reasoning. However, it lacks the capture of fine-grained information of paths. Different from the above methods, our work is the first to integrate fine-grained counterfactual path augmentation information with the reinforcement path reasoning.

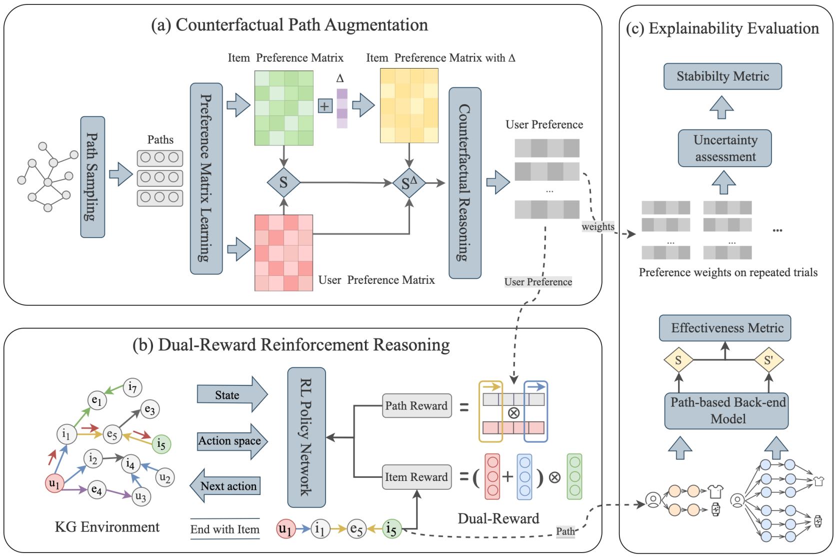

Fig. 2: Overview of Our Framework.

3 Framework of Our Model

Our model's input comprises a set of users $U = \{u_1, u_2...\}$, an item set $I = \{i_1, i_2...\}$ and an interaction set $V_u = \{v|y_{ui} = 1\}$, which contains all observed interactions between u and i. Additionally, we incorporate item attribute information $A = \{a_1, a_2, ...\}$ as entities into the knowledge graph. The knowledge graph G consists of triples $\{(e_h, r, e_t)|e_h, e_t \in E, r \in R\}$, which describes the relation r from the beginning entity e_h, to the tail entity e_t, where E and R denote the set of entities and relations respectively. Given a user u_m, our task is to recommend a set of items $\{i_n\}_{n\in[N]} \subseteq I$, where N is the desired number of recommendations. Furthermore, for each recommended item i_n, we aim to provide a k-hop inference path $p_k(u_m, i_n)$ (where $2 \leq k \leq K$) within the graph G. This path represents a sequence of connected vertices from the user u_m to the item i_n, traversing through k hops (or edges) in the graph.

As illustrated in Fig.2, our model consists of three components. (a) Counterfactual Path Augmentation. We employ counterfactual reasoning to learn fine-grained user preferences pertaining to specific paths. (b) Dual-Reward Reinforcement Reasoning. We utilize a dual-reward mechanism to guide the path reasoning process. (c) Path Explainability Evaluation. We design two novel metrics to evaluate the explainability of models that rely on paths. We will detail these components in Section 4.

4 Our Proposed Model

4.1 Counterfactual Path Augmentation

The component of counterfactual path augmentation integrates counterfactual reasoning techniques, aiming to bolster the analytical capability of path information, thereby precisely discerning which paths and even which relations thereon have significant impacts on rating outcomes. This process essentially involves deeply learning and understanding users' fine-grained preferences. Firstly, we capture users' aspect-level preferences based on the paths between users and items, the relations contained within these paths, and rating data. Subsequently, we learn users' aspect-level preferences exhibited towards specific items. The acquisition of this capability enables us to precisely distinguish users' preference differences for different relations from a fine-grained perspective. We utilize the counterfactual reasoning mechanism to perturb the relations on the paths. Our core objective is to accurately identify the minimal perturbations that exert the greatest negative effects on prediction ratings, as these perturbations correspond to the relations that are crucial in revealing users' preferences.

User Preference Modeling. We abstract the relations embedded within the knowledge graph into aspects that are likely to captivate users' interests. Our objective is to understand how user preferences manifest across various abstract aspects during the recommendation of diverse items to users. Given a user set $U = \{u_1, u_2, \dots, u_m\}$ and an item set $I = \{i_1, i_2, \dots, i_n\}$, we extract all relation types from the knowledge graph, denoted as $R = \{r_1, r_2, \dots, r_r\}$. Consequently, we derive users' aspect-level preference matrix $X \in R_{m \times r}$ and items' aspect-level representation matrix $Y \in R_{n \times r}$. Here, $X_{m,r}$ represents the degree of interest user u_m has in aspect r_r, while $Y_{n,r}$ reflects how well item i_n performs in terms of aspect r_r. X and Y are computed as follows.

$$X_{m,r} = \begin{cases} 0,\ if\ r\ did\ not\ appear\ in\ a\ valid\ path. \\ 1 + (N-1)(\frac{2}{1+exp(-t_{m,r})} - 1), otherwise \end{cases} \tag{1}$$

$$Y_{n,r} = \begin{cases} 0,\ if\ r\ did\ not\ appear\ in\ a\ valid\ path. \\ 1 + \frac{N-1}{1+exp(-t_{n,r} \cdot s_{n,r})}, otherwise \end{cases} \tag{2}$$

Here N represents the rating scale within the system. In this paper, X and Y are derived by computing the frequency of occurrence of each relation for a user u and an item i, respectively. Here, $t_{m,r}$ denotes the frequency of relation r found on valid paths from user u_m to i_n (i.e., items that have been actually clicked on). Similarly, $t_{n,r}$ indicates the frequency of relation r on valid paths originating from item i_n back to user u_m (i.e., users who have purchased the item), and $s_{n,r}$ represents the average frequency. We employ a random walk strategy to identify valid paths within the graph G, initiating from both users and items, and subsequently compute the frequency of relations observed along these valid paths. Both X and Y are then rescaled to fit within the range (1,N) using the Sigmoid function, in order to align with the system's rating scale.

Rating Prediction. Given a user u_m and an item i_n, we utilize a rating prediction function f to predict the rating that u_m would give to i_n, denoted as $s_{m,n} = f(X_m, Y_n|Z, \Theta)$. Here X_m and Y_n are the user vector and item vector, respectively, as defined in Eq.1 and Eq.2. Θ is the model parameters, and Z represents other auxiliary information. In this work, we employ a deep neural network as the underlying architecture for the function f, which comprises a fusion layer followed by three fully connected layers. The network takes as input the concatenated aspect vectors of users and items, and produces a one-dimensional rating $s_{m,n}$ as output. The final output layer utilizes the Sigmoid activation function to map $s_{m,n}$ to the range (0,1). Subsequently, we train the model using cross-entropy loss, as specified in Eq.3. The binary indicator is set to 1 if u_m has previously interacted with i_n, and to 0 otherwise. With this pre-trained model, we can generate predicted ratings for user-item pairs.

$$L = -\sum_{m,n} B_{m,n} \log s_{m,n} + (1 - B_{m,n}) \log(1 - s_{m,n}) \tag{3}$$

Counterfactual Preference Learning. We adopt counterfactual reasoning to identify a user's preference for a specific aspect of an item. By introducing perturbations to the item's aspect vector, if a minor adjustment to a particular aspect results in a significant drop in recommendation performance, it indicates that this aspect is highly significant. Specifically, for a given black-box recommendation model when recommending for the user u_m, we aim to find a minimal perturbation vector $\mathbf{\Delta} = \{\delta_0, \delta_1, \ldots, \delta_r\}$. After applying this perturbation vector $\mathbf{\Delta}$ to the aspect vector Y_n of item i_n, the predicted rating of the recommendation model changes from $s_{u,i}$ to $s_{u,i}^{\mathbf{\Delta}}$. The values within $\mathbf{\Delta}$ are either zero or negative continuous values. We minimize a loss function to learn the optimal perturbation $\mathbf{\Delta}^*$, as illustrated in the following equation:

$$\mathcal{L}_{u,i} = l_1 + \lambda \max(0, \beta + l_2) \tag{4}$$

Here, λ and β serve as hyper-parameters to weigh the terms in the loss function. The loss function comprises two parts: one aimed at ensuring minimal perturbation and the other aimed at maximizing the rating decrease following the perturbation. Firstly, to achieve the smallest possible perturbation, we define the first part of the loss as $l_1 = \|\mathbf{\Delta}\|_2^2 + \alpha \|\mathbf{\Delta}\|_1$, where $\|\cdot\|_1$ and $\|\cdot\|_2$ represent the $L1$ norm and $L2$ norm, respectively. The hyper-parameter α balances the influence of the $L1$ norm and $L2$ norm. The second objective is to maximize the impact on the rating after perturbing the original vector, which can be achieved by minimizing the loss: $l_2 = -s_{u,i} + s_{u,i}^{\mathbf{\Delta}}$.

Finally, after normalizing the learned optimal perturbation $\mathbf{\Delta}^*$, we can derive the preference $C_{ui} = \{p_1, p_2, ..., p_r\}$ of user u towards item i using the Softmax function: $C_{ui} = Softmax(\mathbf{\Delta}_{ui}^*)$. In Section 4.2, we will utilize C_{ui} to calculate a path preference rating, which will serve as one of the dual rewards for enhancing path reasoning.

4.2 Dual-Reward Reinforcement Reasoning

In this section, we propose a reinforcement learning method with dual rewards for generating recommendations alongside their explanations. It integrates path-based rewards with item-focused rewards, aiming to enhance both the precision of the model and its capacity to offer plausible explanations simultaneously.

Formulation as Markov Decision Process. We formalize the graph-based explainable recommendation problem as a Markov Decision Process (MDP). **(1) State.** The state s_t at step t is defined as a tuple (u, e_t, h_t), where $u \in U$ is the starting user entity, e_t is the entity at step t and h_t represents the history of entities visited. The initial state is denoted as $s_0 = (u, u, \emptyset)$. **(2) Action.** The action space A_t for the state s_t is defined as all possible outgoing edges from the entity e_t, i.e., $A_t = \{(r, e)|(e_t, r, e) \in G, e \in \{e_0, \ldots, e_{t-1}\}\}$. **(3) Policy networks.** The policy network π aims to maximize the expected cumulative reward during the path reasoning process. Specifically, π takes the state vector s_t and the action space A_t as inputs, and outputs the probability of each action $p(a_t|s_t, \tilde{A}_u)$. The detailed definition is given in Eq.5. Here $o \in \mathbb{R}^{df}$ means the learned state-hidden feature, $\odot$ denotes the Hadamard product and σ is the nonlinear activation function. The parameters W_1, W_2, W_p are learned by the model. **(4) Optimization.** Our goal is to learn a stochastic policy π that maximizes the expected cumulative reward for any initial user, as described in Eq.6. The discount factor γ and the path preference reward R are crucial components of this optimization problem, which will be discussed next.

$$\begin{aligned} p(a_t|s_t, \tilde{A}_u) = \pi(s_t, \tilde{A}_u) &= softmax(\tilde{A}_u \odot (oW_p)) \\ o &= dropout(\sigma(dropout(\sigma(s_t W_1))W_2)) \end{aligned} \tag{5}$$

$$\mathcal{J}(\theta) = \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \gamma^t R_{t+1} | s_0 = (u, u, \emptyset) \right] \tag{6}$$

Knowledge Graph Embedding. We obtain the embedding of relations and entities in the knowledge graph by maximizing a conditional probability, which is defined as Eq.7. Here $\mathcal{E}$ is the set of entities. The function $f(e_0, e_k|\tilde{r_k})$ is used to measure the similarity between entities e_0 and e_k that are connected by the relation $\tilde{r_k}$. Note that $\tilde{r_k}$ is a valid path for the entity pair (e_0, e_k). We define $f(e_0, e_k|\tilde{r_k})$ in Eq.8. Here $<\cdot, \cdot>$ denotes the dot product operation, and $b_e \in \mathbb{R}$ means the deviation of e.

$$P(e_k|e_0, \tilde{r_k}) = \frac{exp(f(e_0, e_k|\tilde{r_k}))}{\sum_{e' \in \mathcal{E}} exp(f(e_0, e'|\tilde{r_k}))} \tag{7}$$

$$f(e_0, e_k|\tilde{r_k}) =< e_0 + \sum_{s=0}^{k} r_s, e_k > +b_{e_k} \tag{8}$$

Dual-Reward Reinforcement Path Reasoning. In the terminal stage of reinforcement path reasoning, we compute both the matching score reward and the path score reward for the currently reasoned path. This dual-reward mechanism is employed to guide and enhance the agent's reasoning path, ensuring a balance between the credibility and accuracy of the path reasoning model. First, for the matching score reward, at the path reasoning terminal, the reinforcement agent returns a path p_k starting with u and ending with item e_T. We use Eq.8 to calculate the matching score between u and recommended e_T. In this case, $\tilde{r_k}$ represents the relation of *bought*. The formula for the matching score reward R_s is as follows: $R_s = f(u, e_T|\tilde{r_p}) =< u + r_p, e_T > +b_{e_T}$. Then, using the user counterfactual preferences learned in Section 4.1, we augment the path in the reasoning process with fine-grained information. Specifically, at the path inference terminal, the user's path preference is incorporated into the reward of the reinforcement agent. The formula for this is as Eq.9. Here K is the hop count of the path, C_{ui}^k is the preference vector of user u for different relations, and W_u^k is the performance weight of different aspects for u from Eq.1. The reward function for the recommendation step t is given by $R_t = R_s + \xi R_c$, where ξ is a hyper-parameter that balances the two rewards.

$$R_c = \frac{1}{K} \sum_{k=1}^{K} C_{ui}^k \times W_u^k \tag{9}$$

4.3 Evaluation Metrics for Explainability

Assessing whether a model's explanations are credible is subjective.In this paper, we propose two evaluation metrics, namely stability and effectiveness. These two metrics are used to answer the following questions, respectively: (1) Can the model learn stable weights? (2) Can the model identify effective paths?

Stability. Many models rely on learned attention weights to explain themselves. However, numerous studies have found that the reliability of attention mechanisms is weak [2, 3]. If the weights are unstable, the explainable paths will be unclear. The more stable the weights learned by the model, the more reliable its explainability becomes. Therefore, we propose a metric to evaluate the model's ability to learn stable weights. Inspired by information theory [1], we introduce the concept of confidence to quantify the stability of the model. We calculate confidence using entropy, as shown in Eq. 10, which measures uncertainty. The higher the entropy, the lower the confidence, indicating that the model's explainability may be less reliable. Here p_r represents the weight of the r-th aspect learned by the model. Intuitively, better models tend to learn stable weights with lower uncertainty. To compare confidence scores, we will conduct multiple independent repetitions of the experiments (N times), calculate the average entropy of the results, and present these findings in Section 5.

$$Cof = -\sum_r p_r \log(p_r) \tag{10}$$

Effectiveness. The explainability of a model is highly related to the quality of the paths it reasons through. Therefore, we propose an effectiveness metric to assess the model's ability to generate the 'optimal path'. We utilize a path-based recommendation backend model, as shown in Fig.3. The predicted rating S is calculated using the function $S = f(U, I, P, \theta)$, which depends on the user U, item I, path P, and parameters θ. The initial user and item embeddings are obtained using the method described in Section 4.2. For each user who has purchased a set of items $\{i_1^{u_m}, i_2^{u_m}, \dots, i_n^{u_m}\}$, we use a random walk strategy to extract paths between users and items. We then obtain the path embedding by average pooling of entity on the path. The item embedding is updated using the path embedding, i.e., $h^{'}_{i^{u_m}} = f(W_{i^{u_m}} h_{i^{u_m}} + \sum_{z=1}^{Z} W_{p_z} h_{p_z} + b) \odot h_{i^{u_m}})$. Here, f is the ReLU activation function, $h_{i^{u_m}}$ is the initial embedding of i^{u_m}, W and b represent the weight matrix and bias vector, p_z denotes the z-th related path out of Z paths, and $h^{'}_{i^{u_m}}$ is the updated item embedding. Finally, the predicted rating between the user and the item is calculated using a Multilayer Perceptron (MLP) unit, i.e., $S_{i^{u_m}} = MLP(g(h^{'}_{i_1^{u_m}}, h^{'}_{i_2^{u_m}}, \dots, h^{'}_{i_n^{u_m}}))$. Here, g is an aggregation function. The recommendation model is trained using implicit feedback loss through negative sampling. Next, we utilize the well-trained model to evaluate the effectiveness of the paths. Specifically, if an explainable path is meaningful, the user's behavior should exhibit a stronger correlation with that path compared to other paths. Therefore, we only feed the explainable paths learned by the model into the back-end system and examine the new prediction ratings. If the new ratings are very close to the original ratings, it indicates that the generated explainable paths contribute significantly to the analysis of user preferences and can thus be considered more informative. To quantify the effectiveness, we calculate the Mean Squared Error (MSE) between the new ratings and the original ratings, as shown in Eq. 11. Here s_i is the raw rating and $s_i^{'}$ is the rating obtained when only the path output by the model is fed to the back-end. The specific experimental results are shown in Section 5.

$$Efe = \frac{1}{n}\sum_{i=1}^{n}(s_i - s_i^{'})^2 \tag{11}$$

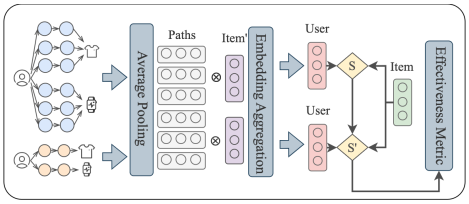

Fig. 3: Evaluation Metrics for Explainability.

5 EXPERIMENT

5.1 Experimental Setup

Datasets. All experiments are conducted on the Amazon e-commerce datasets collection, consisting of product reviews and meta information from Amazon.com. The datasets include four categories: Clothing, Cell Phones, CDs & Vinyl and Beauty. The description of datasets can be found in Table 1.

Table 1: The statistics of datasets.

Datasets	Beauty	Cell Phones	Clothing	CDs & Vinyl
User	22,363	27,879	39,387	75,258
Item	12,101	10,429	23,033	64,443
Feature	22564	22,493	21,366	202,959
Brand	2077	955	1,182	1,414
Category	248	206	1,193	770

Baselines. To evaluate the performance of our model, we compare with the following baseline models. **BPR** [8]: a Bayesian personalized ranking model that learns latent embeddings of users and items. **DeepCoNN** [19]: a text-based convolutional recommendation model which learns user and item representations jointly based on reviews. **JRL** [17]: a joint representation learning model that makes use of multiple sources of information, for instance text and ratings, to train a deep neural network. **CKE** [16]: a representative embedding-based method in which only structural knowledge information is used. **PGPR** [13]: a model that uses reinforcement learning to explore paths and predict preferred items, which is model-specific interpretable recommendation. In order to verify the explainability of the model proposed in this paper, the following models or methods are also selected. **CPER** [5]: a method to improve explainability by leveraging counterfactual at the path level, generates counterfactual weights by perturb the path representation and path topology. **Attention**: a method to distinguish path preferences through an attention mechanism. **Random**: which picks the path randomly.

Implementation Detail. All models are evaluated in terms of three representative top-N recommendation measures: Normalized Discounted Cumulative Gain (NDCG), Recall and Hit Ratio (HR). For the basic black-box model, the learning rate is set to 0.01. For the counterfactual reasoning, λ is set to 5. α is set to 1. For β in the hinge loss, we set $\beta = 0.2$. For the RL module, the state vector has size 400. The discount factor γ is 0.99. For the regulation parameter ξ of the reward function, we set $\xi = 0.1$. For the policy network, $W_1 \in R^{400\times512}, W_2 \in R^{521\times256}, W_p \in R^{256\times250}$. We set a learning rate of 0.0001. The weight of the entropy loss is 0.001. Our code are available at: https://github.com/codeprovided1/CPA-ER

Table 2: Performance comparison for recommendation accuracy.

	Beauty			Cell Phones			Clothing			CDs & Vinyl		
	NDCG	Recall	HR	NDCG	Recall	HR	NDCG	Recall	HR	NDCG	Recall	HR
BPR	2.685	4.192	8.187	1.975	3.142	5.167	0.536	0.981	1.624	1.973	2.481	8.396
DeepCoNN	3.417	5.491	9.869	3.793	6.474	10.125	1.403	2.437	3.397	4.308	6.213	13.965
CKE	3.674	5.824	10.946	3.894	6.915	10.760	1.478	2.469	4.210	4.577	6.582	14.471
JRL	4.119	6.735	12.502	4.191	7.349	10.781	1.563	2.639	4.352	5.015	7.206	16.136
PGPR	5.195	7.979	14.113	5.418	9.117	12.878	2.964	4.977	7.251	5.247	7.351	16.292
CPA-ER	**5.977**	**9.167**	**15.727**	**5.489**	**9.298**	**13.121**	**3.162**	**5.390**	**7.797**	**5.597**	**7.584**	**16.539**

5.2 Overall Performance: RQ1

We compare the proposed recommendation model with five baselines. For each baseline, we report the prediction results in the Table 3. We can find that our recommendation model achieves the best performance on the entire dataset. This means that it is able to hit the true label with a higher probability than other compared methods. Compared to models like PGPR, which suffer from inefficient search strategies and sparse reward signals, our model not only learns users' fine-grained preferences but also generates effective and informative feedback, thereby enhancing the recommendation performance.

5.3 Evaluation of Explainability: RQ2

Stability. We utilized the counterfactual reasoning model proposed in Section 4.1 to learn the preference weights of users and compared with the attention-based baseline model. We randomly selected 100 user-item pairs from the recommendation list and conducted 10 independent repeated experiments. The entropy of the attention weights and counterfactual weights are shown in Fig.4, indicating that our method can learn explanations with lower entropy and greater certainty. We can also see that attention-based models cannot ensure a stable weight distribution through independent runs, which makes the explanation for convincing customers to accept recommendations unpersuasive.

Effectiveness. Here, we compare the effectiveness of paths selected by different path selection methods. Specifically, we compare the proposed counterfactual path augmentation path reasoning method with other baseline models, including random path selection, attention-based methods, and CPER, a path-level counterfactual path weight learning method. To ensure a fair comparison, we keep the number of explainable paths selected by all models consistent. We input the explainable paths learned by the models into the recommendation back-end. By calculating the MSE between the new ratings and the original ratings for all user-item pairs, and display the average values in Fig.5. The results indicate that the paths reasoned by our model contain richer information. We also find that another counterfactual-based model, CPER, performs well, thus proving the effectiveness of counterfactual reasoning in learning weights.

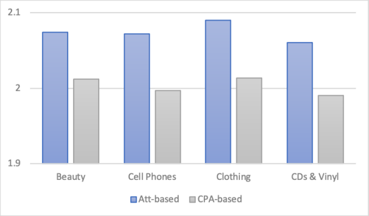

Fig. 4: Stability study.

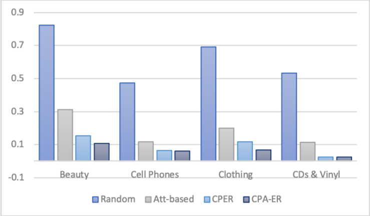

Fig. 5: Effectiveness study.

5.4 Ablation Study: RQ3

The effectiveness of the main components of the model is evaluated next. We verify the change of performance after removing the counterfactual path augmentation. Specifically, No-ER is a method removing counterfactual component. Att-ER is the method that replaces counterfactual model with preferences learned based on attention. CPA-ER is our complete model, which achieves better results on several metrics. As shown in Fig.6 and 7, proving the contribution of the proposed counterfactual module to the accuracy of the model.

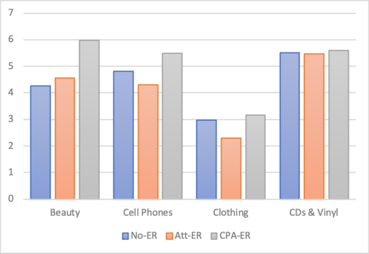

Fig. 6: Ablation study (NDCG).

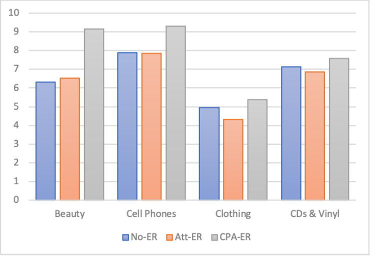

Fig. 7: Ablation study (Recall).

5.5 Effect of Model Hyper-parameters: RQ4

For the reward function adjustment parameter ξ, we set $\{0.05, 0.1, 0.5\}$. According to the Fig.8 and 9, it can be seen that when $\xi = 0.1$, the model has better performance. Because ξ controls the weight of the accuracy and explainability of the recommendation model. When the value of ξ is too large, the model will pay more attention to the explainability of the path and cannot reason more effective items, resulting in the decline of the model performance. However, when ξ is too small, it cannot make full use of the path information learned by counterfactual preference, so it cannot find a more accurate target through the path that users prefer.

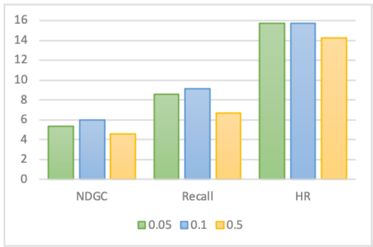

Fig. 8: Performance comparison with different ξ on Beauty

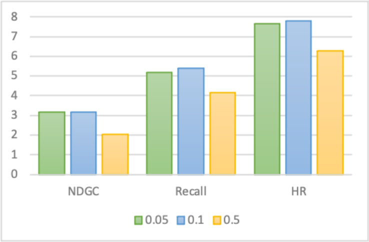

Fig. 9: Performance comparison with different ξ on Cloth

5.6 Case Study: RQ5

We presented a recommendation example to illustrate how the proposed method enhances the explainability of paths. As shown in Fig.10, there are two paths can serve as explanations for recommending the "cream product" to the user. The explanation of $path_1$ and $path_2$ are "it shares the same brand as the facial cleaner you purchased" and "it is described by the same word 'dry skin' with the body lotion you purchased". Additionally, the user previously mentioned the word "dry", and both products the user purchased were described with the word too. It is obvious that the user prefers to buy products suitable for dry skin. Our path augmentation model can generate fine-grained preferences, indicating that when purchasing skincare products, the user cares more about whether the descriptive words match needs rather than the brand. The path scores generated by the model also show that the information $path_2$ better aligns with the user's current purchasing preferences.

6 Conclusion

In this paper, we propose a novel explainable recommendation model incorporating counterfactual path augmentation for reinforcement reasoning. First, we

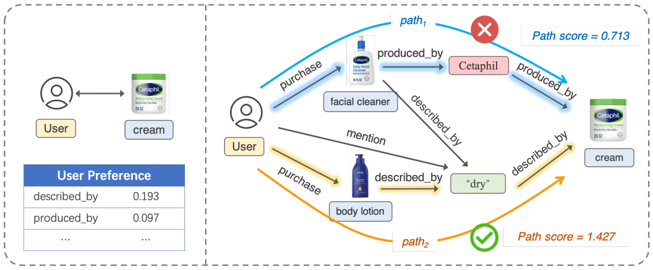

Fig. 10: Real example from Beauty.

propose a user preference learning method based on counterfactual path augmentation, which can precisely distinguish users' preference differences for different relations from a fine-grained perspective. Then we propose a dual-reward reinforcement learning approach to enhance both the accuracy and explainability of the model. We also propose two novel evaluation metrics tailored to assess the quality of explanations. For future work, we will incorporate the review information to enhance explainability. Beyond that, we will explore the deep integration of knowledge graphs and large language models, fully leveraging the rich structured information of knowledge graphs and the powerful language understanding capabilities of large language models.

References

1. L. Brillouin. *Science and information theory*. Courier Corporation, 2013.
2. G. Brunner, Y. Liu, D. Pascual, O. Richter, M. Ciaramita, and R. Wattenhofer. On identifiability in transformers. *arXiv preprint arXiv:1908.04211*, 2019.
3. C. Grimsley, E. Mayfield, and J. R.S. Bursten. Why attention is not explanation: Surgical intervention and causal reasoning about neural models. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1780–1790, Marseille, France, May 2020. European Language Resources Association.
4. A. Li, Y. Zang, Y. Wang, and B. Li. Leveraging interactive paths fornbsp;sequential recommendation. In *Database Systems for Advanced Applications: 28th International Conference, DASFAA 2023, Tianjin, China, April 17–20, 2023, Proceedings, Part II*, page 521–536, Berlin, Heidelberg, 2023. Springer-Verlag.
5. Y. Li, X. Sun, H. Chen, S. Zhang, Y. Yang, and G. Xu. Attention Is Not the Only Choice: Counterfactual Reasoning for Path-Based Explainable Recommendation . *IEEE Transactions on Knowledge & Data Engineering*, 36(09):4458–4471, Sept. 2024.
6. Y. Liu, H. Xuan, B. Li, M. Wang, T. Chen, and H. Yin. Self-supervised dynamic hypergraph recommendation based on hyper-relational knowledge graph. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 1617–1626, New York, NY, USA, 2023. Association for Computing Machinery.

7. N. Ranjbar, S. Momtazi, and M. Homayoonpour. Explaining recommendation system using counterfactual textual explanations. *Mach. Learn.*, 113(4):1989–2012, Sept. 2023.
8. S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.
9. C.-Y. Tai, L.-Y. Huang, C.-K. Huang, and L.-W. Ku. 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, page 879–889, New York, NY, USA, 2021. Association for Computing Machinery.
10. J. Tan, S. Xu, Y. Ge, Y. Li, X. Chen, and Y. Zhang. Counterfactual explainable recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 1784–1793, New York, NY, USA, 2021. Association for Computing Machinery.
11. S. Verma, V. Boonsanong, M. Hoang, K. Hines, J. Dickerson, and C. Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Comput. Surv.*, 56(12), Oct. 2024.
12. X. Wang, K. Liu, D. Wang, L. Wu, Y. Fu, and X. Xie. Multi-level recommendation reasoning over knowledge graphs with reinforcement learning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2098–2108, New York, NY, USA, 2022. Association for Computing Machinery.
13. Y. Xian, Z. Fu, S. Muthukrishnan, G. de Melo, and Y. Zhang. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 285–294, New York, NY, USA, 2019. Association for Computing Machinery.
14. K. Xiong, W. Ye, X. Chen, Y. Zhang, W. X. Zhao, B. Hu, Z. Zhang, and J. Zhou. Counterfactual review-based recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 2231–2240, New York, NY, USA, 2021. Association for Computing Machinery.
15. S. Xu, J. Xu, B. Li, and X. Fu. Predicting where you visit innbsp;anbsp;surrounding city: A mobility knowledge transfer framework based onnbsp;cross-city travelers. In *Database Systems for Advanced Applications: 28th International Conference, DASFAA 2023, Tianjin, China, April 17–20, 2023, Proceedings, Part I*, page 334–350, Berlin, Heidelberg, 2023. Springer-Verlag.
16. F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362, 2016.
17. Y. Zhang, Q. Ai, X. Chen, and W. B. Croft. Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1449–1458, 2017.
18. K. Zhao, X. Wang, Y. Zhang, L. Zhao, Z. Liu, C. Xing, and X. Xie. Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 239–248, New York, NY, USA, 2020. Association for Computing Machinery.
19. L. Zheng, V. Noroozi, and P. S. Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 425–434, 2017.