



Intrinsically motivated reinforcement learning based recommendation with counterfactual data augmentation

Xiaocong Chen¹ · Siyu Wang¹ · Lianyong Qi² · Yong Li³ · Lina Yao^{1,4}

Received: 16 January 2023 / Revised: 28 April 2023 / Accepted: 6 June 2023 /
Published online: 15 July 2023
© The Author(s) 2023

Abstract

Deep reinforcement learning (DRL) has shown promising results in modeling dynamic user preferences in RS in recent literature. However, training a DRL agent in the sparse RS environment poses a significant challenge. This is because the agent must balance between exploring informative user-item interaction trajectories and using existing trajectories for policy learning, a known exploration and exploitation trade-off. This trade-off greatly affects the recommendation performance when the environment is sparse. In DRL-based RS, balancing exploration and exploitation is even more challenging as the agent needs to deeply explore informative trajectories and efficiently exploit them in the context of RS. To address this issue, we propose a novel intrinsically motivated reinforcement learning (IMRL) method that enhances the agent’s capability to explore informative interaction trajectories in the sparse environment. We further enrich these trajectories via an adaptive counterfactual augmentation strategy with a customised threshold to improve their efficiency in exploitation. Our approach is evaluated on six offline datasets and three online simulation platforms, demonstrating its superiority over existing state-of-the-art methods. The extensive experiments show that our IMRL method outperforms other methods in terms of recommendation performance in the sparse RS environment.

Keywords Recommender systems · Deep reinforcement learning · Counterfactual reasoning

1 Introduction

Recently, deep reinforcement learning (DRL) has received increasing interest in RS due to its capability in capturing users’ dynamic interests [1]. Current DRL-based RS can be generally categorized into three streams: value-based methods, policy-based methods, and hybrid methods. One of the representatives of value-based methods would be Deep Q-learning (DQN), which [2] brought into news recommendation. However, deep Q-learning-based methods require the “maximize” operation over the action space (i.e., all the candidate items), which is not traceable and may induce agent stuck problem [3]. Policy-gradient methods can

✉ Xiaocong Chen
xiaocong.chen@unsw.edu.au

Extended author information available on the last page of the article

mitigate such problems but suffer from the high variance problem as the optimization is based on the last step's trajectory, which could be distinct from previous trajectories [4]. The hybrid method is a combination of policy-gradient and value-based methods. It aims to reduce the variance for policy-gradient by introducing the value-based method [5] and has gained more attention [6–9].

However, the user-item interactions are commonly sparse, hindering policy optimizations to find rewards via exploration as well as maximizing performance via exploitation. Specifically, as DRL relies on carefully engineering environment rewards that are extrinsic to the agents, the sparsity barely provides dense reward signals (i.e., most of the reward signals might be missing because of the highly incomplete interactions and user feedback). Hence, new exploration strategies would be a choice to encourage agents to discover a wider range of states and formulate richer interaction trajectories [10]. Recent literature [11, 12] shows that exploration is effective in reducing model uncertainty in regions of sparse rewards or user interactions. Moreover, most of the existing works in DRL RS apply ϵ -greedy as the exploration strategy, where the agent has ϵ possibility to conduct exploration randomly [1]. However, random exploration increases training time and uncertainty and may not be able to explore enough informative interaction trajectories. Moreover, it will cost a considerable number of trials, making it infeasible to apply to highly sparse user feedback in recommender systems, as it also requires a significant number of trials for exploitation, which is known as the exploration and exploitation trade-off.

Differently, several attempts have been made from a different perspective of data augmentation to relieve the sparsity. Experience replay is widely used in DRL methods, empowering agents to learn by reusing past interaction trajectories. However, experience replay can only promote certain trajectories to be replayed [13]. The policy learning process may be harmed if the generated trajectories are not informative. Recent studies also investigate equipping data augmentation with causality by governing to generate informative trajectories. For instance [14], designs a simple counterfactual method by measuring the embedding change to generate a new user sequence.

Moreover [15], consider the embedding to contain two parts, which are dispensable or indispensable items related to the final recommended items, by leveraging causality. By replacing dispensable items, it can generate more user sequences but with the same performance. The main limitation with these approaches, however, is that they assume an embedding is fixed and never changes, while the embedding should be dynamic and updated after each interaction. Moreover, the agent never knows the ground truth (i.e., the user's final choice) during online interactions. Hence, it is impractical to leverage the ground truth to determine the embedding difference or indispensable items as existing works have done.

It should be noted that although various causal data augmentation techniques have been successfully implemented for sequential or traditional recommendation problems, none have yet been applied to DRL-based methods. In DRL, the agent learns policies from previously collected trajectories, but the collection process is often a costly bottleneck. Given the potential benefits of causal data augmentation in recommender systems, it is worth exploring its applicability to DRL methods. However, the existing methods cannot be directly transferred to DRL, as they are not designed for the same learning paradigm. Two fundamental reasons for this are: (i) existing methods focus on embeddings, whereas the traditional DRL method does not involve any embedding or similar concept. However, some works on RS consider the state representation as an embedding to construct the entire pipeline. This approach requires an individual representation network, which can be computationally inefficient and expensive as extra hyperparameters are introduced. , and (ii) existing methods assume embeddings are

constant and do not change, whereas user interests in DRL are dynamic and may shift after the system provides recommendations or over time.

Furthermore, most of the existing DRL-based methods utilize random exploration [1], which may not be well-suited for recommendation tasks. This is due to two reasons: i). firstly, traditional DRL algorithms are intended for countable state spaces with limited potential states. However, in recommender systems, the state space is uncountable [1], and random exploration may fail to reach useful states in certain episodes. ii). Moreover, random exploration can generate a multitude of uninformative trajectories, which are stored in the replay buffer and could impede the training process. DRL methods heavily rely on the replay buffer to acquire knowledge from previous interactions [16].

In order to address the above issues, we propose a new end-to-end model, namely Intrinsically Motivated Reinforcement Learning with Counterfactual Augmentation (IMRL), from two aspects: augmenting informative trajectories and a new exploration strategy. We design a novel empowerment-based exploration strategy to encourage the agent to explore potentially informative interaction trajectories in the sparse environment. Moreover, we elaborate on a new counterfactual data augmentation method for DRL RS to augment those newly explored informative trajectories so that they can have a higher exposure probability, thus boosting the final performance.

In summary, we make the following contributions in this paper:

- We propose a novel DRL method, IMRL, which can augment trajectories that are causally valid but never seen by the agent to alleviate the data sparsity problem. Moreover, we also introduce an adaptive threshold to dynamically control the boundary of the informative trajectories as the learning process in DRL is evolutionary.
- We designed an empowerment-driven exploration strategy for IMRL to help explore those unexplored but potentially informative interaction trajectories. Our experiments show that the designed exploration strategy can boost the final performance in the online simulation platforms.
- We have conducted extensive experiments in both offline and online settings and shown the superiority of IMRL. We conducted offline experiments with six well-known datasets and online experiments in three public simulation platforms.

Differences with our previous work [17]. Building upon our previous work [17], we present a deeper investigation of the mechanism underlying our proposed data augmentation method in this study. Specifically, we introduce a novel threshold parameter $T_{\max}$ to enable adaptive control over the data augmentation process, as we have discovered that informative trajectories depend on the training process. Additionally, we provide a comprehensive analysis of each component of our proposed method. To further validate the efficacy of our approach, we conduct experiments on two additional offline datasets and two online environments.

2 Background

In this section, we will provide some background on the proposed work, which can be divided into two parts: DRL RS problem formulation and causality. Firstly, we will briefly introduce the problem formulation using MDP. Secondly, we will introduce local causal models as an extension of the commonly used structural causal models.

2.1 Problem formulation

Reinforcement learning-based recommender systems learn from interactions through a Markov Decision Process (MDP). Given a recommendation problem consisting of a set of users $\mathcal{U} = u_0, u_1, \dots, u_n$, a set of items $\mathcal{I} = i_0, i_1, \dots, i_m$, and users' demographic information $\mathcal{D} = d_0, d_1, \dots, d_n$, MDP can be represented as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where each represents the following:

- $\mathcal{S}$ denotes the state space, which is the combination of the subsets of $\mathcal{I}$ and $\mathcal{D}$. It represents the user's previous interactions and demographic information. Based on that, it can be written in a compositional form: $\mathcal{S} = \mathcal{S}^1 \oplus \mathcal{S}^2 \oplus \dots \oplus \mathcal{S}^n$ for a fixed n , which represents the dynamic count of components [18];
- $\mathcal{A}$ is the action space, which represents the agent's selection during recommendation based on the state space $\mathcal{S}$. Similarly, it can also be written in a compositional form: $\mathcal{A} = \mathcal{A}^1 \oplus \mathcal{A}^2 \oplus \dots \oplus \mathcal{A}^n$;
- $\mathcal{P}$ is the set of transition probabilities for state transfer based on the action received, which also refers to users' behavior probabilities. It is worth mentioning that $\mathcal{P}$ will not be estimated in this study as we are using a model-free reinforcement learning approach;
- $\mathcal{R}$ is a set of rewards received from users, which are used to evaluate the action taken by the recommendation system, with each reward being a binary value to indicate user's click; γ is a discount factor;
- $\gamma \in [0, 1]$ for balancing the future reward and current reward.

Given a user u_0 and the state s_0 observed by the agent (or the recommendation system), which includes a subset of the item set $\mathcal{I}$ and user's demographic information d_0 , a typical recommendation iteration for user u_0 goes as follows: First, the agent makes an action a_0 based on the recommend policy π_0 under the observed initial state s_0 and receives the corresponding reward r_0 . Then, the agent generates a new policy π_1 based on the received reward r_0 and determines the new state s_1 based on the probability distribution $p(s_{new}|s_0, a_0) \in \mathcal{P}$. The cumulative reward after k iterations is as follows (Figure 1):

$$r_c = \sum_{k=0}^{\infty} \gamma^k r_k.$$

2.2 Local causal models

Structural Causal Models (SCMs) [19] can be represented as a tuple: $\mathcal{M}_t(V_t, U_t, F)$ with the following components based on the state and action composition form at timestamp t . It is normally represented as a directed acyclic graph (DAG) $\mathcal{G}$ with the following components:

- $V_t = s_t^1, s_t^2, \dots, s_t^n, a_t^1, \dots, a_t^m, s_{t+1}^1, \dots, s_{t+1}^n$, which represents the nodes in DAG $\mathcal{G}$.
- $U_t = u^1, \dots, u^{2n+m}$ is a set of noise variables, one for each node in V_t . It is determined by the initial state, past actions, and environment. We assume that the noise variable is time-independent, which implies that $U_t \perp\!\!\!\perp U_{t+1}$.
- F is a set of functions that map $U_t \times \text{Parentage}(V_t) \rightarrow V_t$, where $\text{Parentage}(\cdot)$ means the parent node of $\cdot$.

We assume that the dynamic count of states is n and the dynamic count of actions is m . The state observed at timestamp t is written as s_t , which is the composition of $s_t^1, s_t^2, \dots, s_t^n$. Local causal models are an extension of SCMs that only consider the local causal effect [20].

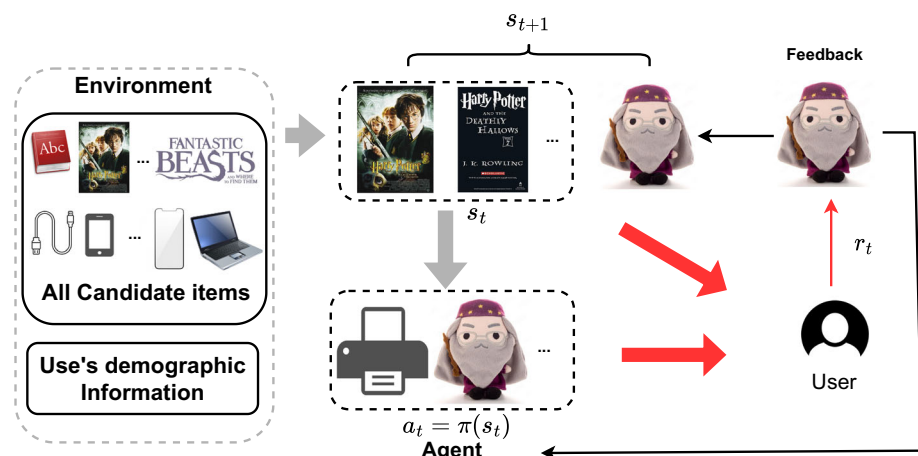


Figure 1 Illustration of a typical interaction between a user and a DRL agent in DRL RS. The red line represents the user's information flow, and the grey line represents the recommender's information flow. The states are sampled from the environment

A local causal model can be represented as $\mathcal{M}_t^{\mathcal{L}}(V_t^{\mathcal{L}}, U_t^{\mathcal{L}}, F^{\mathcal{L}})$, with the DAG $\mathcal{G}^{\mathcal{L}}$ derived from the global causal model $\mathcal{M}_t$ in the subspace $\mathcal{L}$, having the same components with the following additional constraints:

$$\text{Parentage}(V_t^{\mathcal{L}}) = \text{Parentage}(V_t | (s_t, a_t) \in \mathcal{L}), \quad (1)$$

$$\text{Parentage}(U_t^{\mathcal{L}}) = \text{Parentage}(U_t | (s_t, a_t) \in \mathcal{L}). \quad (2)$$

Moreover, the local causal model requires the set of edges in $\mathcal{G}$ to be structurally minimal [21].

3 Methodology

In this section, we will briefly explain the proposed approach for reinforcement learning-based recommendation, Intrinsically Motivated Reinforcement Learning with Counterfactual Augmentation (IMRL), which can address the sparse interactions problem in DRL RS. We are addressing this problem from two aspects based on the aforementioned challenges: i) using a novel adaptive data augmentation method to generate more potentially informative interaction trajectories by employing counterfactual reasoning; and ii) designing a new exploration strategy by introducing an intrinsic reward signal to encourage the agent to explore. In contrast to our conference version, this study offers a more thorough exploration of the mechanism that underlies our proposed data augmentation method. Our focus is on introducing a novel threshold parameter, denoted as $T_{\max}$, which enables adaptive control over the data augmentation process. This parameter is crucial because we have found that informative trajectories are dependent on the training process.

Hence, the proposed IMRL consists two main components: *Counterfactual reasoning for data augmentation* and *Intrinsically Motivated Exploration*.

3.1 Counterfactual data augmentation

Formally, given an arbitrary trajectory $\tau : (s, a, r, s')$ sampled from the replay buffer, where r is the reward signal received by the agent when action a is executed at state s , most of the trajectories in the large candidate item set situation are not informative, resulting in zero rewards. As a result, it is challenging to sample informative trajectories since the number of non-informative trajectories is significantly larger. Augmenting these informative trajectories to increase the likelihood of sampling them is a straightforward solution. We assume that the state s_{t+1} satisfy the SCM:

$$s_{t+1} = f(s_t, a_t, U_{t+1}), \quad (3)$$

$f(\cdot)$ represents the causal mechanism, a_t is the action taken at timestamp t , and U_{t+1} is the noise term that is independent of (s_t, a_t) . Our main objective is to estimate the causal mechanism $f(\cdot)$ and generate more data that is unseen but causally valid. However, estimating the global $f(\cdot)$ is challenging [22]. To overcome this challenge, we draw inspiration from recent advances in local causal models [20, 23] and focus on estimating the local causal mechanism $f_l(\cdot)$. The local causal model assumes the existence of a local directed acyclic graph (DAG) $\mathcal{G}^{\mathcal{L}(i \perp j)}$ in a subspace $\mathcal{L}^{i \perp j}$ such that $\mathcal{L}^{i \perp j} \subset \mathcal{L}$ where $\mathcal{L} : \mathcal{S} \times \mathcal{A}$. It satisfies the following condition:

$$s_{t+1}^j \perp\!\!\!\perp V_t^i | \text{Parentage}(s_{t+1}^j) \setminus V_t^i, (s_t, a_t) \in \mathcal{L}^{j \perp i}, \quad (4)$$

where $\perp\!\!\!\perp$ is used to represent the independence. In recommender systems, there is a large subspace of states in which users' previous interests will not affect the final recommendation, as users' interests are dynamic. By focusing on the subspace $(s_t, a_t) \in \mathcal{L}^{(j \perp i)}$, we can formulate a local causal model $\mathcal{M}^{\mathcal{L}^{(j \perp i)}}$ such that the local DAG $\mathcal{G}^{\mathcal{L}^{(j \perp i)}}$ contains no edge from V_t^i to s_{t+1}^j . This implies that the local DAG $\mathcal{G}^{\mathcal{L}^{(j \perp i)}}$ is strictly sparser than the global DAG $\mathcal{G}$. With this property, we can use the local causal model to conduct data augmentation to alleviate the sparsity problem in DRL RS.

Consider the counterfactual question “What if user u had been interested in item j instead of item i at timestamp t ?” This question can be described in causal form as *What if component s_t^i had the value x instead of y at timestamp t ?* It can be solved by using Pearl's do-calculus to the causal model $\mathcal{M}$ to obtain a sub-model,

$$\mathcal{M}_{\text{do}(s_t^i=x)}^{\mathcal{L}} = (V, U, F_x) \text{ where } F_x = F \setminus f^i \cup s_t^i = x. \quad (5)$$

Moreover, the incoming edge to s_t^i will be removed from $\mathcal{G}_{\text{do}(s_t^i=x)}$. Now, we utilize the local causal model to generate data that is unseen by the agent but causally valid. In order to achieve this, we will augment the data based on the counterfactual modification with the subset of causal factors at timestamp t and keep the remaining factors unchanged. Such an augmentation process can use the counterfactual model $\mathcal{M}_{\text{do}(s_t^i=x)}^{\mathcal{L}}$ to modify the causal factors $s_t^{i \cdots j}$ and regenerate the corresponding children in the DAG.

However, such a process is computationally expensive in our recommendation scenario as it requires re-sampling for the children in the DAG. Inspired by the idea of collaborative filtering and the state composition form which was mentioned previously, we can simplify the process by omitting the sampling step. Specifically, the core of the augmentation is to estimate $\mathcal{M}_{\text{do}(s_t^i=x)}^{\mathcal{L}}$, which can be obtained easily by assuming that similar users will have similar interests, an idea inherent in collaborative filtering. Under such an assumption, we can obtain $\mathcal{M}_{\text{do}(s_t^i=x)}^{\mathcal{L}}$ by replacing the causally independent component of s_t using

the local causal model. For example, we can identify those interaction histories that do not affect the current recommendation, and are thus causally independent of the current action a_t . The overall algorithm¹ can be found in Algorithm 1.

Algorithm 1: Counterfactual Data Augmentation

```

input: Two trajectories  $\tau_1$  and  $\tau_2$  from replay buffer  $D$ 
1 function Augmentation ( $\tau_1, \tau_2$ ) is
2    $s_1, a_1, s'_1 \leftarrow \tau_1, s_2, a_2, s'_2 \leftarrow \tau_2$ ;
3    $m_1, m_2 \leftarrow \text{ind}(s_1, a_1), \text{ind}(s_2, a_2)$ ;
4    $d \leftarrow \text{random sample from } (m_1 \cap m_2)$ ;
5    $s_g, a_g, s'_g = s_1, a_1, s'_1.\text{copy}()$ ;
6    $s_g[d], a_g[d], s'_g[d] = s_2[d], a_2[d], s'_2[d]$ ;
7    $\tau_g \leftarrow s_g, a_g, s'_g, R(s_g, a_g)$ ;
8    $D.\text{append}(\tau_g)$  if  $s_g[d], a_g[d], s'_g[d]$  exists else  $D$ ;
9   return  $D$ ;
10 end
11 function ind( $s, a$ ) is
12   Generate the adjacency matrix  $M$  by using  $s$  and  $a$ ;
13   Find the connected (dependent) components set  $C$ ;
14   return index of independent components in  $\mathcal{G} \setminus C$ ;
15 end
  
```

3.2 Intrinsically motivated exploration

The second aspect we use to address sparsity is intrinsically motivated exploration strategies. We propose to use empowerment to represent intrinsic motivation, which can boost the agent's exploration capability, allowing it to reach more states and produce corresponding potentially informative interaction trajectories.

Empowerment is an information-theoretic method in which an agent executes a sequence of k actions $\mathbf{a}^k \in \mathcal{A}$ while in state $s \in \mathcal{S}$, according to an exploration policy $\pi_{\text{empower}}(s, \mathbf{a}^k)$ (which we will shorten to $\pi_e(s, \mathbf{a}^k)$). This exploration policy is a conditional probability distribution: $\pi_e : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$. The agent's goal is to identify an optimal policy π_e that maximizes the mutual information $I[\mathbf{a}^k, s'|s]$ between the action sequence $\mathbf{a}^k$ and the state s' to which the environment transitions after executing the sequence $\mathbf{a}$ in the current state s . This can be formulated as follows:

$$\mathbb{E}(s) = \max_{\pi_e} I[\mathbf{a}^k, s'|s] \quad (6)$$

$$= \max_{\pi_e} \mathbb{E}_{\pi_e(s, \mathbf{a}^k) \mathcal{P}(s, \mathbf{a}^k, s')} \log \left[\frac{p(\mathbf{a}^k, s'|s)}{\pi_e(\mathbf{a}^k, s)} \right]. \quad (7)$$

The aim is to maximize the expectation of the logarithmic ratio between the joint probability distribution of the action sequence $\mathbf{a}^k$, the state s' , and the state transition probability distribution $\mathcal{P}(s, \mathbf{a}^k, s')$, and the exploration policy $\pi_e(\mathbf{a}^k, s)$. By maximizing this quantity, the agent can boost its exploration capability and reach more states, which can in turn produce potentially informative interaction trajectories.

¹ For set-based representations

Here, $\mathbb{E}(s)$ refers to the optimal empowerment value, and $\mathcal{P}(s, \mathbf{a}^k, s')$ refers to the probability of transitioning to s' after executing the action sequence $\mathbf{a}^k$ in state s , where $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$. Importantly,

$$p(\mathbf{a}^k, s' | s) = \frac{\mathcal{P}(s, \mathbf{a}^k, s') \pi_e(\mathbf{a}^k, s)}{\sum_{\mathbf{a}^{k'}} \mathcal{P}(s, \mathbf{a}^{k'}, s') \pi_e(\mathbf{a}^{k'}, s)} \quad (8)$$

is the inverse dynamics model of π_e . The optimal empowerment values are obtained by the policy π^* that maximizes $\mathbb{E}\pi^*(s)$.

However, the above definition of empowerment is more general than the RL setting since it considers a k -step policy, while RL usually considers a single-step policy. Moreover, estimating the k -step empowerment is challenging. Therefore, in this study, we use $k = 1$ to narrow down the empowerment into the RL setting, which only considers one step ahead. The Blahut-Arimoto algorithm [24, 25] shows that empowerment can be solved in low-dimensional discrete settings. Additionally [26], uses parametric function approximators to estimate empowerment in high-dimensional and continuous state-action spaces. It provides theoretical guarantees for using empowerment in recommender systems since state-action spaces are high-dimensional [1]. There are two possibilities for utilizing empowerment in RL:

- Find high mutual information between actions and the subsequent state achieved by that action.
- Train a behavioral policy to take an action in each state such that the expected empowerment value of the next state is highest.

Both approaches are feasible for the normal reinforcement learning setting, which can encourage the agent to take an action that can result in the maximum number of future states. However, there is some conceptual difference between them. The second approach seeks states with a large number of reachable next states [27, 28], while the first approach aims to find high mutual information between actions and subsequent states, which is not necessarily the same as seeking highly empowered states [26]. The first approach can be achieved by transforming the state and its subsequent states' representations into KL divergence and minimizing it [29]. However, this transformation introduces extra complexity and information loss, which may affect performance. The second approach, which uses the behavioral policy to explore highly empowered states, would be more suitable and simple for our setting. The main reason is that we are using a model-free approach to solve the problem. Model-free RL methods maintain two policies, which are the target policy π and the behavior policy π_e . The second approach is more suitable for model-free approaches as it does not require extra computational cost to traverse all subsequent states and calculate the KL divergence. It can be easily adopted into existing RL frameworks.

Hence, the goal of the MDP process with the empowerment can be rewritten as,

$$\max_{\pi_b} \mathbb{E}_{\pi_b, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t (\alpha \cdot R(s_t, a_t) + \beta \cdot \frac{p(a_t | s_{t+1}, s_t)}{\pi_b(a_t, s_t)}) \right] \quad (9)$$

π_b is the behavior policy, and α and β are constants used to balance the weight of instant reward and empowerment. We include the empowerment term as an additional component to the reward signal $R(s_t, a_t)$ in order to encourage exploration by the agent.

3.3 Training procedure

From an information-theoretic perspective, optimizing for empowerment is equivalent to optimizing the inverse dynamics [28, 30] based on the distribution $\pi_e(s, a)$. Therefore, we introduce the inverse dynamics into the objective function to calculate the empowerment. Our method is built on Soft Actor-Critic (SAC) [31] with temperature tuning and deterministic policy. The overall training algorithm can be found in Algorithm 2.

We follow the same training strategy as the standard SAC algorithm. However, since the empowerment is introduced, we modify the objective function to ensure that the empowerment term can be optimized. We use several function approximators to learn different components in the proposed method. The value function V is parameterized by ψ , the Q-function is parameterized by θ , the target policy is parameterized by ϕ , and the inverse dynamics is parameterized by ξ . Since we are using an off-policy algorithm where the transition probability is not learned, we use $\mathcal{P}$ to represent the state transition probability in the environment. The soft Q-function can be trained by minimizing the following objective function:

$$J_Q(\theta) = \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}} [Q_\theta(s_t, a_t) - (r(s_t, a_t) + \gamma V_\psi(s_{t+1}))^2]. \quad (10)$$

The target function V_ψ can be optimized by minimizing:

$$J_V(\psi) = \mathbb{E}_{s_t \sim \mathcal{D}} \left[V_\psi(s_t) - \mathbb{E}_{a_t \sim \pi_\phi} \left(Q_\theta(s_t, a_t) + \underbrace{\beta g(s_t, a_t)}_{\text{policy}} \right)^2 \right], \quad (11)$$

where β is a constant used to balance the empowerment. Different from the origin SAC algorithm, we replace the policy term from $-\log \pi_\phi(s_t, a_t)$ to $g(s_t, a_t)$ to consider the empowerment where $g(s_t, a_t)$ is defined as:

$$g(s_t, a_t) = \mathbb{E}_{\mathcal{P}(s' | s_t, a_t)} [\log p_\xi(a_t | s', s_t) - \log \pi_\phi(s_t, a_t)]. \quad (12)$$

Note that, different from s_t , the s' represent all the possible subsequent states where a_t is executed in state s_t at timestamp t .

Similarly, the optimization of the policy $\pi(\phi)$ can be written as:

$$J_\pi(\phi) = -\mathbb{E}_{s_t \sim \mathcal{D}} \left[\mathbb{E}_{a_t \sim \pi_\phi} [\beta g(s_t, a_t) + Q_\theta(s_t, a_t)] \right], \quad (13)$$

where apply the same substitution. The inverse dynamic $p(\xi)$ will be updated based on:

$$J_p(\xi) = -\mathbb{E}_{\pi_\phi} [\log p_\xi(a_t | s', s_t)]. \quad (14)$$

Lastly, the temperature parameter will be adjusted automatically by using the following entropy method [32]:

$$J(\alpha) = -\mathbb{E}_{a_t \sim \pi_\phi} [\alpha \log \pi_\phi(s_t, a_t) + \alpha \mathcal{H}]. \quad (15)$$

Note that, SAC uses exponentially averaged value ψ' to stabilize the training process [33]. The update rule can be written as: $\psi' \leftarrow \lambda_{\psi'} \psi + (1 - \lambda_{\psi'}) \psi'$.

Moreover, we perform data augmentation on the replay buffer after each interaction to generate causally valid, unseen trajectories. This augmentation can provide more trajectories

Algorithm 2: Overall training algorithm

```

1 Initialize Parameters  $\phi$  for policy,  $\theta$  for Q-networks,  $\psi$  for value function,  $\xi$  for inverse dynamic,
  replay buffer  $\mathcal{D}$ , threshold  $T$ . ;
2 for each episode till converge do
3    $s_0 \leftarrow$  initiate state ;
4   for each environment step do
5      $a_t \sim \pi_\phi(a_t | s_t)$ ;
6      $r_t \leftarrow \mathcal{R}(s_t, a_t)$ ;
7      $s_{t+1} \leftarrow \mathcal{P}(s_{t+1} | s_t, a_t)$ ;           /* Sample from environment */
8      $\mathcal{D} \leftarrow \mathcal{D} \cup \{s_t, a_t, r_t, s_{t+1}\}$ ;
9     if  $r_t \geq T$  then
10      Augmentation( $\mathcal{D}$ ) ;
11    end
12    Update  $T$  based on (16);
13  end
14  for each gradient step do
15     $\theta \leftarrow \theta - \lambda_Q \nabla_\theta J_Q(\theta)$  ;
16     $\psi \leftarrow \psi - \lambda_\psi \nabla_\psi J_V(\psi)$ ;
17     $\phi \leftarrow \phi - \lambda_\phi \nabla_\phi J_\pi(\phi)$ ;
18     $\xi \leftarrow \xi - \lambda_\xi \nabla_\xi J_P(\xi)$ ;
19     $\alpha \leftarrow \alpha - \lambda_\alpha \nabla_\alpha J(\alpha)$  ;
20  end
21 end

```

at the early stage to increase the number of samples. Specifically, most of the model parameters are learned by sampling from the replay buffer $\mathcal{D}$. The training process can be described as searching for states or state-action pairs in $\mathcal{D}$ to update the target policy such that the received reward is maximized. As the augmentation introduces more samples into the replay buffer, the gradient update process has a higher chance of achieving a better policy.

It is worth mentioning that we only augment informative trajectories. However, the definition of informative trajectories highly depends on the learning progress. We believe that every trajectory with non-zero reward is informative in the early stages but harmful when the final target policy is close to optimal. Hence, we selectively conduct augmentation with the replay buffer to ensure that zero-reward trajectories are not augmented to increase sparsity. However, as the interaction progresses, the way we determine informative trajectories changes. Some trajectories are informative in the early stages as the agent needs to explore all possibilities. In the later stages, the agent will pursue higher rewarding trajectories, making those low-rewarding trajectories less useful. In such situations, we design an adaptive threshold to evaluate whether the trajectory is worth augmenting or not. The designed adaptive threshold is intuitive, where a moving average is used. It can be represented as:

$$T = \sigma / \lambda_d \text{ if } T \leq T_{max} \text{ else } T_{max} \quad (16)$$

σ is a custom constant used to determine the initial value of the threshold, and the decay rate $\lambda_d \in (0, 1]$ decreases as the number of episodes increases. By setting (σ, λ_d) to appropriate values, we can achieve a monotonically increasing threshold. Ideally, we start with the initial values $\sigma = 1$ and $\lambda_d = 1.1$. T_{max} is a constant specific to the environment that represents the maximum reward that the agent can achieve at each step.

4 Experiments

In this section, we conduct experiments to answer three main research questions:

- **RQ1:** Does IMRL outperform existing DRL approaches in both offline and online settings?
- **RQ2:** Can IMRL help to alleviate the sparsity of interactions problem in DRL RS in online simulation environments?
- **RQ3:** How does each component contribute to the final performance in online simulation environments?

In contrast to our conference version, we offer a thorough examination of each aspect of our proposed technique. In order to substantiate the effectiveness of our approach, we perform experiments on two more offline datasets and two online environments.

4.1 Experiment setup

In order to demonstrate the superiority of IMRL, we conducted experiments in both offline and online simulation settings.

4.1.1 Offline datasets

We use six publicly available datasets:

- **MoveLens-20M**² is a dataset about the user behavior of watching movies.
- **Librarything**³ is a dataset about book review information.
- **Book-crossing**⁴ is a dataset related to book preference.
- **Netflix Prize**⁵ is a dataset from Netflix yearly competition for recommendation.
- **Amazon-CD**⁶ is e-commerce datasets which contains user's purchase behavior.
- **GoodReads**⁷ is a book dataset.

The statistics of those datasets are summarized in Table 1.

Moreover, due to the unique interaction logic in reinforcement learning-based methods, an additional data preparation process is required to ensure that the agent can interact with offline datasets. We adopt the same strategy as in previous work [34] to convert those datasets into reinforcement learning environments so that IMRL can interact with them.

4.1.2 Baselines and offline evaluation metrics

We selected the following baselines, which include both non-reinforcement learning based methods and reinforcement learning based methods:

- SASRec [35], a well-known baseline for sequential recommendation methods that utilize the self-attention mechanism.

² <https://grouplens.org/datasets/movielens/20m/>

³ http://cseweb.ucsd.edu/~jmcauley/datasets.html#social_data

⁴ <http://www2.informatik.uni-freiburg.de/~chiegler/BX/>

⁵ <https://www.kaggle.com/netflix-inc/netflix-prize-data>

⁶ <https://jmcauley.ucsd.edu/data/amazon/>

⁷ <https://www.goodreads.com/>

Table 1 Statistics of the datasets used in our offline experiments

Dataset	# of Users	# of Items	# of Inter.	Density
Amazon CD	75,258	64,443	3,749,004	0.08%
Librarything	73,882	337,561	979,053	0.004%
Book-Crossing	278,858	271,379	1,149,780	0.0041%
GoodReads	808,749	1,561,465	225,394,930	0.02%
MovieLens-20M	138,493	27,278	20,000,263	0.53%
Netflix	480,189	17,770	100,498,277	1.18%

- CASR [14], a counterfactual data augmentation method for sequential recommendation. As CASR only conducts the augmentation, we selected STAMP [36] to make recommendations, which is described in CASR.
- CauseRec [15], a counterfactual sequence generation method for sequential recommendation.
- CoCoRec [37], a category-aware collaborative method for sequential recommendation.
- CGKR [38], a counterfactual generation method for alleviating spurious correlations.
- DEERS [36], a reinforcement learning-based recommendation method that considers both positive and negative feedback.
- KGRL [6], a reinforcement learning-based method that utilizes the capability of GCN to process the knowledge graph information.
- TPGR [7], a model that uses reinforcement learning and binary trees for the large-scale interactive recommendation.
- PGPR [39], a knowledge-aware model that employs reinforcement learning for explainable recommendation.

It is worth mentioning that because of the different training paradigms of these two kinds of methods (i.e., supervised learning and reinforcement learning), we cannot guarantee that the comparison with existing non-reinforcement learning-based state-of-the-art methods is strictly fair. We conducted the supervised learning-based methods in the same setting as well as the reinforcement learning-based methods.

We used the same training and hyperparameter settings as in [15] for the non-DRL based methods. We used Adam as the main optimizer, with an embedding size of 32 and a batch size of 1024. For IMRL, we use 100, 000 episodes and a batch size of 1024. For DRL-based methods, we trained all models with 100, 000 episodes on VirtualTB, but only 1, 000 episodes on RecoGym and RecSim. As reported in the original paper, we set all hyperparameters to their default values. For the proposed model, T_{max} was set to 10 for VirtualTB, and 1 for RecoGym and RecSim. The learning rate was set to 0.001. Recall, precision, nDCG are selected as the evaluation metrics. And they are reported based on top-20 recommendation. It should be noted that IMRL achieves lower precision scores than the baselines on LibraryThing and Book-Crossing datasets. The slight decrease of 0.25% in precision on LibraryThing is not surprising and could be attributed to the randomness of the model, which is still considered acceptable. However, for Book-Crossing, IMRL shows a significant drop in precision compared to CauseRec. This could be due to the high sparsity of the Book-Crossing dataset, which may require more episodes to collect trajectories or a strong exploration strategy which can be left as the future direction. It is important to consider that supervised learning and reinforcement learning have different learning paradigms, making it difficult to control the number of episodes required.

4.1.3 Online simulation

Unlike offline datasets, online simulation platforms are based on gym⁸, which is a standard toolkit for reinforcement learning research. We conducted online experiments on three widely used public simulation platforms that mimic online recommendations in real-world applications: VirtualTB [40], RecSim [41], and RecoGym [42].

VirtualTB is a real-time simulation platform for recommendation, in which the agent recommends items based on users' dynamic interests. It uses a pre-trained generative adversarial imitation learning (GAIL) model to generate different users who have both static and dynamic interests. The interactions between users and items are also generated by the GAIL model. This allows VirtualTB to provide a large number of users and corresponding interactions to simulate real-world scenarios. After initialization, VirtualTB generates different users each time, and the dynamic interests of each user change after each interaction.

RecSim is a platform for creating configurable simulation environments that support sequential interactions between users and recommender systems. Unlike VirtualTB, RecSim has fewer users and items but offers a range of simpler tasks. For our experiments, we chose to use the "interest evolution" task, which encourages the agent to explore and satisfy the user's interests without further exploitation.

RecoGym is a smaller platform where users do not have long-term goals. Unlike RecSim and VirtualTB, RecoGym is designed for computational advertising. Similar to RecSim, RecoGym uses clicks or non-clicks to represent the reward signal. Additionally, like RecSim, users in these two environments do not have any dynamic interests.

4.1.4 Baselines for online simulation

In our online simulation experiments, all the baselines are based on reinforcement learning. Therefore, non-reinforcement learning methods are excluded as they cannot interact with gym-based environments. It is worth mentioning that some methods require additional side information from the environment that is not present in these three platforms. Therefore, we had to remove those components to ensure a fair comparison, where every method received the same state representation. The primary evaluation metric used for online simulation is Click-Through-Rate (CTR), which is determined by the platform.

IMRL is implemented using Pytorch [43]. All experiments are conducted on a server with two Intel Xeon E5-2697 v2 CPUs, 4 NVIDIA TITAN X Pascal GPUs, 2 NVIDIA TITAN RTX GPUs, 2 NVIDIA RTX A5000 GPUs, and 768 GB of memory. We provide details about the model parameters for reproducibility. The hidden units for both the actor and critic networks are set to 256. The learning rate, discount factor, and size of the replay buffer are set to 0.0003, 0.99, and 1e6, respectively, during the experiments. For VirtualTB, the training episode is set to 1e6, and testing is conducted every 10 episodes. For RecoGym and RecSim, the training episode is set to 10,000, and testing is conducted every 10 episodes.

4.2 Offline experiments

The complete results can be found in Table 2. We found that our method IMRL generally outperforms all existing state-of-the-art methods, including both non-reinforcement learning-

⁸ <https://gym.openai.com>

Table 2 The overall results of our model comparison with several state-of-the-art models in different datasets

Dataset Measure (%)	Amazon CD		Librarything		nDCG
	Recall	Precision	Recall	Precision	
SASRec	5.129 ± 0.233	2.349 ± 0.144	8.419 ± 0.294	6.726 ± 0.139	7.471 ± 0.221
CASR	8.321 ± 0.212	5.012 ± 0.129*	14.213 ± 0.311	12.512 ± 0.219	13.555 ± 0.198
CauseRec	9.124 ± 0.213*	4.892 ± 0.299	14.222 ± 0.421	12.582 ± 0.321	12.875 ± 0.317
CoCoRec	8.982 ± 0.221	4.982 ± 0.312	13.982 ± 0.123	11.233 ± 0.300	12.098 ± 0.302
CGKR	9.172 ± 0.182	4.981 ± 0.142	14.523 ± 0.382	12.642 ± 0.192	13.462 ± 0.205
DEERS	7.123±0.221	3.581 ± 0.200	10.422 ± 0.231	10.321 ± 0.355	11.872 ± 0.241
KGRL	8.208 ± 0.241	4.782 ± 0.341	12.128 ± 0.241	12.451 ± 0.242	13.925 ± 0.252*
TPGR	7.294 ± 0.312	2.872 ± 0.531	14.713 ± 0.644*	12.410 ± 0.612	13.225 ± 0.722
PGPR	6.619 ± 0.123	1.892 ± 0.143	11.531 ± 0.241	10.333 ± 0.341	12.641 ± 0.442
Ours	9.213 ± 0.219	5.032 ± 0.125	14.829 ± 0.321	12.610 ± 0.231*	14.021 ± 0.335
Improvement	0.45 %	1.02%	2.11%	-0.25%	0.69%
Book-Crossing					
SASRec	5.831 ± 0.272	3.184 ± 0.149	GoodReads		
CASR	8.322 ± 0.300	5.012 ± 0.211	6.921 ± 0.312	5.242 ± 0.211	6.124 ± 0.210
CauseRec	9.213 ± 0.213	6.213 ± 0.198	11.228 ± 0.123	10.922 ± 0.339	10.233 ± 0.210
CoCoRec	8.234 ± 0.231	5.182 ± 0.200	11.827 ± 0.431*	10.982 ± 0.412	10.277 ± 0.312
CGKR	9.242 ± 0.197*	5.234 ± 0.183	10.882 ± 0.233	10.012 ± 0.210	10.012 ± 0.129
DEERS	7.321 ± 0.320	2.574 ± 0.201	11.577 ± 0.290	10.878 ± 0.287	10.428 ± 0.226*
KGRL	8.004 ± 0.223	3.521 ± 0.332	8.231 ± 0.122	9.318 ± 0.132	9.401 ± 0.184
TPGR	7.246 ± 0.321	4.523 ± 0.442	7.459 ± 0.401	11.444 ± 0.321*	10.331 ± 0.331
PGPR	6.998 ± 0.112	3.932 ± 0.121	11.219 ± 0.323	10.322 ± 0.442	9.825 ± 0.642
Ours	9.331 ± 0.213	5.442 ± 0.124*	11.421 ± 0.223	10.042 ± 0.212	9.234 ± 0.242
Improvement	0.96%	-12.4%	12.013 ± 0.201	11.726 ± 0.138	10.612 ± 0.320
			1.57%	2.46%	1.76%

Table 2 continued

Dataset Measure (%)	Amazon CD		Librarything		nDCG
	Recall	Precision	Recall	Precision	
	MovieLens-20M		Netflix		
SASRec	14.512 ± 0.510	12.412 ± 0.333	11.321 ± 0.231	10.322 ± 0.294	14.225 ± 0.421
CASR	17.324 ± 0.212	14.021 ± 0.210	13.551 ± 0.240	12.412 ± 0.122	15.212 ± 0.211
CauseRec	17.625 ± 0.331	14.982 ± 0.291	13.982 ± 0.325*	12.842 ± 0.222*	15.882 ± 0.261
CoCoRec	16.212 ± 0.211	14.222 ± 0.290	13.762 ± 0.199	12.001 ± 0.129	13.284 ± 0.235
CGKR	17.672 ± 0.255*	14.986 ± 0.266*	13.427 ± 0.286	12.752 ± 0.177	15.478 ± 0.223*
DEERS	16.123 ± 0.312	12.984 ± 0.221	12.847 ± 0.219	11.321 ± 0.294	14.521 ± 0.401
KGRL	16.021 ± 0.498	14.989 ± 0.432	13.009 ± 0.343	11.874 ± 0.232	13.082 ± 0.348
TPGR	16.431 ± 0.369	13.421 ± 0.257	12.512 ± 0.556	11.512 ± 0.595	10.425 ± 0.602
PGPR	14.234 ± 0.207	9.531 ± 0.219	10.982 ± 0.181	10.123 ± 0.227	10.134 ± 0.243
Ours	17.798 ± 0.231	15.041 ± 0.122	14.421 ± 0.239	13.012 ± 0.321	15.448 ± 0.122
Improvement	0.71 %	0.37%	3.14%	1.33%	-2.72%

The result was reported based on top-20 recommendation and the highest results are in bold and the second highest is marked by *

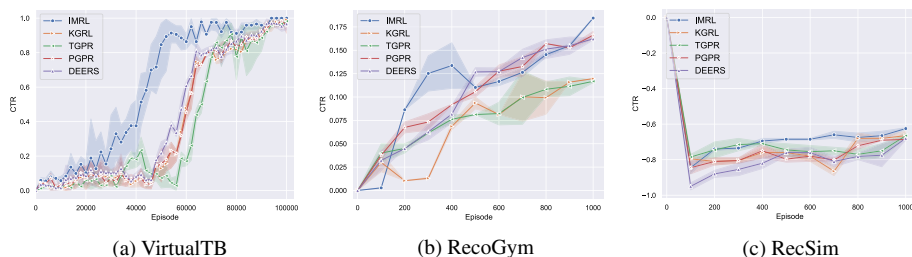


Figure 2 Overall results for online simulation environments

based methods and reinforcement learning-based methods. It should be noted that IMRL does not outperform CauseRec in two datasets, but it still performs better than all other methods. Although IMRL has lower precision than CauseRec in Book-Crossing, we observe that the recall and nDCG are better than CauseRec. A similar situation occurs in Netflix, where the nDCG of IMRL is lower than CauseRec, but precision and recall are better than CauseRec.

4.3 Online experiments (RQ2)

We also report the performance of the selected reinforcement learning-based baselines in three online simulation environments. The results can be found in Figure 2. As we can see, IMRL outperforms all the others in all of the selected three simulation platforms. The performance in RecoGym and RecSim is quite close as those two environments are very small and do not require a complex exploration policy. Hence, we will focus on the later discussion in VirtualTB as it has a more complex environment which is more similar to the real-world situation.

The simplest way to evaluate the sparsity is to measure the speed of the model that tends to converge. In reinforcement learning, the way we use to measure sparsity is the number of useful samples that are fed into the agent via the replay buffer or sampled from the environment. Hence, dense environments can boost the model to converge at an early stage. In Figure 2a, we can see that IMRL has an outstanding speed of convergence than other methods in VirtualTB, which shows that it can overcome the sparse environment. In RecoGym and RecSim, IMRL also demonstrates considerable improvement when compared with those baselines. The main reason is that RecoGym and RecSim are small environments that contain only a few items and users. The sparsity is not serious and can be handled by random exploration.

4.4 Ablation study (RQ3)

To answer RQ3, we conducted experiments with the two major components of IMRL: empowerment and augmentation. The results of this study can be found in Figure 3, where IMRL-E denotes IMRL without empowerment, and IMRL-A denotes IMRL without augmentation. Additionally, we investigated the effect of the different strategies of empowerment in IMRL, including the KL-divergence approach mentioned in Section 3.2. We use IMRL-KL to represent this method.

We observed that both components play an important role in IMRL and contribute jointly to its final performance. Furthermore, we noticed that IMRL-KL did not perform as well

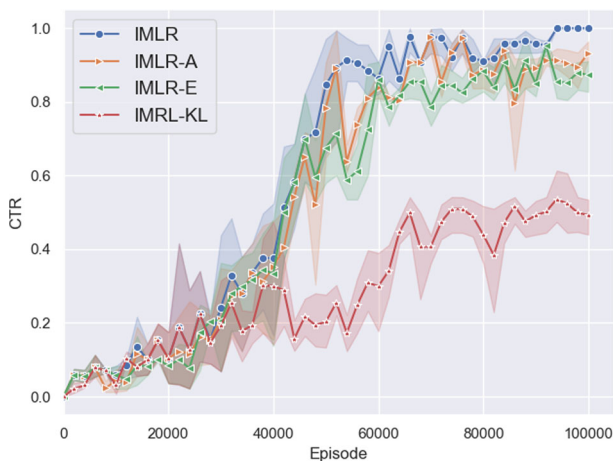


Figure 3 Ablation study on VirtualTB

as the other methods. One possible reason for this is that information is lost during the transformation in the calculation of KL-divergence. Hence, we can infer that our approach of using empowerment is better than KL-Divergence. In the next part, we will investigate the effect of the adaptive threshold.

We observed that removing empowerment-based exploration resulted in a drop in the model's performance. As we mentioned earlier, empowerment-based exploration has a higher probability of reaching or producing informative states, which can enhance the model's performance. This indicates that random exploration has limitations and may harm the model's performance in recommendation tasks. However, we did not observe a significant improvement with the augmentation component, only a slight one. One possible reason for this is that VirtualTB is a simulation platform that can simulate real-world situations, but the number of states is limited due to computational resource constraints. A larger simulator may reveal a more pronounced difference in performance between augmented and non-augmented models. However, further study is required, and it is not the goal of this paper.

4.5 Impact of the adaptive augmentation

An important difference between our previous work [17] and this study is the role played by the adaptive augmentation threshold in early-stage discovery and reducing the number of uninformative trajectories. In this section, we investigate how the threshold affects early-stage performance. We start with $\sigma = 10$ and $\lambda_d = 1.1$. Note that the decay function of λ_d varies depending on the environment. We use VirtualTB as the primary evaluation platform and the following decay function:

$$\lambda_d \leftarrow \lambda_d - \left\lceil \frac{\# \text{ of episodes}}{100,000} \right\rceil,$$

with $T_{max} = 10$. We report the CTR at different stages of IMRL with adaptive threshold compared to IMRL without adaptive threshold (referred to as IMRL-T for short) in Table 3. We repeated the experiments five times with five different random seeds and report the average value. We found that with the adaptive threshold, IMRL can reach peak performance

Table 3 The effect of the adaptive threshold in VirtualTB

Method	Episodes('000)									
	10	20	30	40	50	60	70	80	90	100
IMLR-T	0.05	0.14	0.23	0.38	0.57	0.66	0.72	0.82	0.89	0.94
IMLR	0.11	0.20	0.38	0.56	0.88	0.80	0.92	0.90	0.92	0.98

around 70, 000 episodes, whereas, without the adaptive augmentation threshold, it takes until 90, 000 episodes to reach peak performance. However, when the number of episodes reaches 100, 000, the performance of both methods is similar, which supports our theory that adaptive augmentation can improve the early-stage performance of the model.

5 Related work

In this section, we will briefly review two topics related to our work: reinforcement learning-based recommendation and causality in recommender systems.

Reinforcement learning-based recommendation Reinforcement learning (RL) has been used in recommendation systems (RS) to provide personalized recommendations. Zheng et al. [2] introduced deep RL into RS using the Deep Q-Network (DQN) to recommend news articles. Double DQN was used to build a user's profile, and an activeness score was designed to evaluate whether the user is active or not. Zhao et al. [36] extended this method by introducing negative feedback. Chen et al. [34] used cascading DQN and a generative user model method to handle unknown reward situations. Chen et al. [44] introduced a scalable policy-gradient-based method for recommendation by introducing a policy correction gradient estimator to reduce the variance, and [4] designed a Pairwise Policy Gradient method to reduce variance. Chen et al. [7] proposed a tree-based method for large-scale interactive recommendation using the actor-critic algorithm. [6] integrated the knowledge graph into the actor-critic structure and used graph convolutional networks to capture information. Xian et al. [39] designed a knowledge graph-based environment for explainable recommendations. Chen et al. [9] focused on reward function design and used inverse RL to avoid elaborate reward functions in the online recommendation.

Causality in recommender systems : Causality has become a popular research topic in recent literature on recommendation systems due to its wide usage in debiasing and data augmentation for RS. For example, [45] employed model-agnostic counterfactual reasoning to address popularity bias in RS, while [46] proposed causal intervention. Conversely, [15] separated users' historical actions into dispensable and indispensable items and generated new user sequences by replacing dispensable items. Causality has shown a strong connection with RL in recent years as both can affect the input's status [47]. Zhu et al. [48] employed actor-critic algorithms to discover different Directed Acyclic Graph (DAG) structures for causal discovery. Dasgupta et al. [49] proposed a meta-RL framework to conduct causal reasoning by exploring different causal structures. The causal inference has also been used to determine unobserved confounders to improve the performance of imitation learning [50], and [51] utilized causal inference to build an explainable RL model. Moreover, recent works [52, 53] are focusing on using causality to enhance interpretability and debias.

Our contributions We propose a novel end-to-end model called Intrinsically Motivated Reinforcement Learning with Counterfactual Augmentation (IMRL), which focuses on two key aspects: enhancing informative trajectories and introducing a new exploration strategy. Our approach incorporates a unique empowerment-based exploration strategy that motivates the agent to explore informative interaction trajectories in a sparse environment. Additionally, we introduce a new counterfactual data augmentation technique for Deep Reinforcement Learning Reward Shaping (DRL RS), which amplifies the exposure probability of these newly discovered informative trajectories, thereby improving the overall performance of the model.

6 Conclusion

In this paper, we propose IMRL to address the sparse interaction problem in DRL-based RS from two perspectives: quantity and quality. We propose a counterfactual-based method to augment informative interaction trajectories and an empowerment-based exploration to boost the possibility of finding high-quality trajectories. We conducted experiments on both offline datasets and online simulation platforms to demonstrate the superiority of the proposed method.

In the future, we plan to explore the potential of empowerment and develop novel solutions to address the sparse interaction problem in DRL-based RS. Additionally, one of the limitations of IMRL is the sub-optimality problem. Although the proposed adaptive method can enhance performance at the initial stage, it may result in sub-optimal trajectories being augmented. To address this, we aim to design a more detailed adaptive strategy, rather than a conventional one, to further improve the performance of the proposed model in the future.

Author Contributions X.Chen: Conceptualization, Methodology, Writing - Original. Draft S.Wang: Conceptualization, Writing - Review & Editing. L. Qi: Writing - Review & Editing. Y. Li: Writing - Review & Editing. L. Yao: Writing - Supervision, Conceptualization, Review & Editing.

Funding Not Applicable

Availability of data and materials All the datasets are public available, and the links are attached in the footnotes.

Declarations

Competing interests None

Ethical Approval Not Applicable

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Chen, X., Yao, L., McAuley, J., Zhou, G., Wang, X.: Deep reinforcement learning in recommender systems: A survey and new perspectives. *Knowl. Based Syst.* **264**, 110335 (2023)
- Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N.J., Xie, X., Li, Z.: Drn: A deep reinforcement learning framework for news recommendation. In: *Proceedings of the 2018 World Wide Web Conference*, 167–176 (2018)
- Dulac-Arnold, G., Evans, R., van Hasselt, H., Sunehag, P., Lillicrap, T., Hunt, J., Mann, T., Weber, T., Degris, T., Coppin, B.: Deep reinforcement learning in large discrete action spaces. [arXiv:1512.07679](https://arxiv.org/abs/1512.07679) (2015)
- Xu, J., Wei, Z., Xia, L., Lan, Y., Yin, D., Cheng, X., Wen, J.-R.: Reinforcement learning to rank with pairwise policy gradient. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 509–518 (2020)
- Degris, T., White, M., Sutton, R.S.: Off-policy actor-critic. [arXiv:1205.4839](https://arxiv.org/abs/1205.4839) (2012)
- Chen, X., Huang, C., Yao, L., Wang, X., Zhang, W., et al: Knowledge-guided deep reinforcement learning for interactive recommendation. In: *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8 (2020). IEEE
- Chen, H., Dai, X., Cai, H., Zhang, W., Wang, X., Tang, R., Zhang, Y., Yu, Y.: Large-scale interactive recommendation with tree-structured policy gradient. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 3312–3320 (2019)
- Cai, Q., Filos-Ratsikas, A., Tang, P., Zhang, Y.: Reinforcement mechanism design for e-commerce. In: *Proceedings of the 2018 World Wide Web Conference*, 1339–1348 (2018)
- Chen, X., Yao, L., Sun, A., Wang, X., Xu, X., Zhu, L.: Generative inverse deep reinforcement learning for online recommendation. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 201–210 (2021)
- Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Overcoming exploration in reinforcement learning with demonstrations. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6292–6299 (2018). IEEE
- Chen, M., Wang, Y., Xu, C., Le, Y., Sharma, M., Richardson, L., Wu S.-L., Chi, E.: Values of user exploration in recommender systems. In: *Fifteenth ACM Conference on Recommender Systems*, 85–95 (2021)
- Chen, M.: Exploration in recommender systems. In: *Fifteenth ACM Conference on Recommender Systems*, pp. 551–553 (2021)
- Schaul, T., Quan, J., Antonoglou, I., Silver, D. (2015) Prioritized experience replay. [arXiv:1511.05952](https://arxiv.org/abs/1511.05952)
- Wang, Z., Zhang, J., Xu, H., Chen, X., Zhang, Y., Zhao, W.X., Wen, J.-R.: Counterfactual data-augmented sequential recommendation. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 347–356 (2021)
- Zhang, S., Yao, D., Zhao, Z., Chua, T.-S., Wu, F.: Causerec: Counterfactual user sequence synthesis for sequential recommendation. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 367–377 (2021)
- Chen, X., Yao, L., McAuley, J., Guan, W., Chang, X., Wang, X.: Localitysensitive state-guided experience replay optimization for sparse rewards in online recommendation. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1316–1325 (2022)
- Chen, X., Yao, L., Chang, X., Wang, S.: Empowerment-driven policy gradient learning with counterfactual augmentation in recommender systems. In: *2022 IEEE International Conference on Data Mining (ICDM)*, 885–890 (2022). IEEE
- Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., Smola, A.: Deep sets. [arXiv:1703.06114](https://arxiv.org/abs/1703.06114) (2017)
- Pearl, J.: Causal inference in statistics: An overview. *Stat. Surv.* **3**, 96–146 (2009)
- Pitis, S., Creager, E., Garg, A.: Counterfactual data augmentation using locally factored dynamics. [arXiv:2007.02863](https://arxiv.org/abs/2007.02863) (2020)
- Peters, J., Janzing, D., Schölkopf, B.: *Elements of Causal Inference - Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, MA, USA (2017)
- Lu, C., Huang, B., Wang, K., Hernández-Lobato, J.M., Zhang, K., Schölkopf, B.: Sample-efficient reinforcement learning via counterfactual-based data augmentation. [arXiv:2012.09092](https://arxiv.org/abs/2012.09092) (2020)
- Xiang, Y., Truong, M.: Acquisition of causal models for local distributions in bayesian networks. *IEEE Trans. Cybern.* **44**(9), 1591–1604 (2013)
- Arimoto, S.: An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Trans. Inf. Theory* **18**(1), 14–20 (1972)

25. Blahut, R.: Computation of channel capacity and rate-distortion functions. *IEEE Trans. Inf. Theory* **18**(4), 460–473 (1972)
26. Mohamed, S., Jimenez Rezende, D.: Variational information maximisation for intrinsically motivated reinforcement learning. *Adv. Neural Inf. Process. Syst.* **28**, 2125–2133 (2015)
27. Jung, T., Polani, D., Stone, P.: Empowerment for continuous agent-environment systems. *Adapt. Behav.* **19**(1), 16–39 (2011)
28. Leibfried, F., Pascual-Díaz, S., Grau-Moya, J.: A unified bellman optimality principle combining reward maximization and empowerment. *Adv. Neural Inf. Process. Syst.* **32**, 7869–7880 (2019)
29. Kumar, N.M.: Empowerment-driven exploration using mutual information estimation. [arXiv:1810.05533](https://arxiv.org/abs/1810.05533) (2018)
30. Elements of Information Theory. John Wiley & Sons, Ltd. <https://doi.org/10.1002/0471200611>
31. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: *International Conference on Machine Learning*, pp. 1861–1870 (2018). PMLR
32. Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al.: Soft actor-critic algorithms and applications. [arXiv:1812.05905](https://arxiv.org/abs/1812.05905) (2018)
33. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. *Nature* **518**(7540), 529–533 (2015)
34. Chen, X., Li, S., Li, H., Jiang, S., Qi, Y., Song, L.: Generative adversarial user model for reinforcement learning based recommendation system. In: *International Conference on Machine Learning*, pp. 1052–1061 (2019). PMLR
35. Kang, W.-C., McAuley, J.: Self-attentive sequential recommendation. In: *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206 (2018). IEEE
36. Zhao, X., Zhang, L., Ding, Z., Xia, L., Tang, J., Yin, D.: Recommendations with negative feedback via pairwise deep reinforcement learning. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1040–1048 (2018)
37. Cai, R., Wu, J., San, A., Wang, C., Wang, H.: Category-aware collaborative sequential recommendation. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 388–397 (2021)
38. Mu, S., Li, Y., Zhao, W.X., Wang, J., Ding, B., Wen, J.-R.: Alleviating spurious correlations in knowledge-aware recommendations through counterfactual generator. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1401–1411 (2022)
39. Xian, Y., Fu, Z., Muthukrishnan, S., de Melo, G., Zhang, Y.: Reinforcement knowledge graph reasoning for explainable recommendation. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 285–294 (2019). ACM
40. Shi, J.-C., Yu, Y., Da, Q., Chen, S.-Y., Zeng, A.-X.: Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4902–4909 (2019)
41. Ie, E., Hsu, C.-w., Mladenov, M., Jain, V., Narvekar, S., Wang, J., Wu, R., Boutilier, C.: Recsim: A configurable simulation platform for recommender systems. [arXiv:1909.04847](https://arxiv.org/abs/1909.04847) (2019) [cs.LG]IMRL
42. Rohde, D., Bonner, S., Dunlop, T., Vasile, F., Karatzoglou, A.: Recogym: A reinforcement learning environment for the problem of product recommendation in online advertising. [arXiv:1808.00720](https://arxiv.org/abs/1808.00720) (2018)
43. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems*, pp. 8026–8037 (2019)
44. Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., Chi, E.H.: Top-k off-policy correction for a reinforce recommender system. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 456–464 (2019)
45. Wei, T., Feng, F., Chen, J., Wu, Z., Yi, J., He, X.: Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1791–1800 (2021)
46. Zhang, Y., Feng, F., He, X., Wei, T., Song, C., Ling, G., Zhang, Y.: Causal intervention for leveraging popularity bias in recommendation. [arXiv:2105.06067](https://arxiv.org/abs/2105.06067) (2021)
47. Gershman, S.J.: Reinforcement learning and causal models. *The Oxford handbook of causal reasoning*, 295 (2017)
48. Zhu, S., Ng, I., Chen, Z.: Causal discovery with reinforcement learning. [arXiv:1906.04477](https://arxiv.org/abs/1906.04477) (2019)
49. Dasgupta, I., Wang, J., Chiappa, S., Mitrovic, J., Ortega, P., Raposo, D., Hughes, E., Battaglia, P., Botvinick, M., Kurth-Nelson, Z.: Causal reasoning from meta-reinforcement learning. [arXiv:1901.08162](https://arxiv.org/abs/1901.08162) (2019)

50. Zhang, J., Kumor, D., Bareinboim, E.: Causal imitation learning with unobserved confounders. *Advances in neural information processing systems* 33 (2020)
51. Madumal, P., Miller, T., Sonenberg, L., Vetere, F.: Explainable reinforcement learning through a causal lens. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 2493–2500 (2020)
52. Ji, J., Li, Z., Xu, S., Xiong, M., Tan, J., Ge, Y., Wang, H., Zhang, Y.: Counterfactual collaborative reasoning. In: *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pp. 249–257 (2023)
53. He, X., Zhang, Y., Feng, F., Song, C., Yi, L., Ling, G., Zhang, Y.: Addressing confounding feature issue for causal recommendation. *ACM Trans. Info. Syst.* **41**(3), 1–23 (2023)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Xiaocong Chen¹ · Siyu Wang¹ · Lianyong Qi² · Yong Li³ · Lina Yao^{1,4}

Siyu Wang
siyu.wang5@unsw.edu.au

Lianyong Qi
lianyongqi@gmail.com

Yong Li
liyong07@tsinghua.edu.cn

Lina Yao
lina.yao@unsw.edu.au

¹ School of Computer Science and Engineering, University of New South Wales, Sydney 2052, NSW, Australia

² College of Computer Science and Technology, China University of Petroleum (East China), Dong Ying Shi, China

³ Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

⁴ Data 61, CSIRO, Eveleigh 2015, NSW, Australia