Contrastive Representation for Interactive Recommendation

Jingyu Li, Zhiyong Feng, Dongxiao He, Hongqi Chen, Qinghang Gao, Guoli Wu

College of Intelligence and Computing, Tianjin University
{lijingyu_working, zyfeng, hedongxiao, hongqi, gaoqh, wuguoli_it999}@tju.edu.cn

Abstract

Interactive Recommendation (IR) has gained significant attention recently for its capability to quickly capture dynamic interest and optimize both short and long term objectives. IR agents are typically implemented through Deep Reinforcement Learning (DRL), because DRL is inherently compatible with the dynamic nature of IR. However, DRL is currently not perfect for IR. Due to the large action space and sample inefficiency problem, training DRL recommender agents is challenging. The key point is that useful features cannot be extracted as high-quality representations for the recommender agent to optimize its policy. To tackle this problem, we propose Contrastive Representation for Interactive Recommendation (CRIR). CRIR efficiently extracts latent, high-level preference ranking features from explicit interaction, and leverages the features to enhance users' representation. Specifically, the CRIR provides representation through one representation network, and refines it through our proposed Preference Ranking Contrastive Learning (PRCL). The key insight of PRCL is that it can perform contrastive learning without relying on computations involving high-level representations or large potential action sets. Furthermore, we also propose a data exploiting mechanism and an agent training mechanism to better adapt CRIR to the DRL backbone. Extensive experiments have been carried out to show our method's superior improvement on the sample efficiency while training an DRL-based IR agent.

1 Introduction

Interactive Recommendation (IR) is recently popular and becoming accepted as an reasonable recommender workflow. Traditionally, the recommendation problem was considered to be a classification or prediction task (such us collaborative filtering and content-based filtering methods). However, this may not match the real recommendation scenario. It is now widely agreed that formulating it as a sequential decision problem can better reflect the user-system interaction (Lin et al. 2023). Therefore, IR can be formulated as a Markov decision process and be solved by Reinforcement Learning (RL) or Deep Reinforcement Learning (DRL). DRL-based IR can naturally capture users' unique dynamic interests and balance between short and long term targets, similar to the well-known Reinforcement Learning based on Human Feedback (RLHF) mechanism in ChatGPT (OpenAI 2024). There has been a variety of commercial services on interactive recommendation systems based on DRL (Chen et al. 2019b; Yu, Shen, and Jin 2019; Zhou et al. 2020; Cai et al. 2023a).

However, sample inefficiency is a significant issue that hinders the further development of IR (Yu 2018). Sample efficiency refers to the training performance which can be achieved limited to a certain number of training samples. It measures the training difficulty of an RL agent. For IR tasks, the DRL models usually turn out to be even more sample-inefficient than other typical DRL tasks (e.g., robotic control, game agents, etc.) (Chen et al. 2021). Because conducting DRL from high dimensional observations is empirically observed to be sample-inefficient (Lake et al. 2017; Kaiser et al. 2024). Unfortunately, IR usually has to encode users' profiles into high dimensional observations to convey abundant semantic information. This makes the IR agents can be hardly trained to the ideal effect within limited online interaction. So it cannot quickly attract users' interest, leading to the failure of maintaining a certain number of active users (Gao et al. 2023b). This is a fatal problem for the online recommendation business.

Some approaches have been proposed to address the sample efficiency problem in IR. Commonly, they can be classified into three streams of methods based on different intention (some methods may belong to more than one category): **(i)** Improve functional components in DRL; **(ii)** Increase significative reward signals. **(iii)** Enhancing the state representation method. The first class enhance the policy for making action (Zou et al. 2020) or the way of exploiting samples (Chen et al. 2022b). The second class usually trains offline user simulator to simulate users' behaviours and give reward feedback towards recommendation (Shi et al. 2019; Ie and other 2019; Rohde et al. 2018; Zhao et al. 2023). The last class aims at enhancing the representation methods for extracting users' profile (Liu et al. 2020; Xi et al. 2023). Works of the last class are usually based on the consensus (Laskin, Srinivas, and Abbeel 2020): *If an agent can acquire high quality semantic information from high dimensional observations, DRL-based recommendation methods built on top of those representations should be significantly more sample-efficient*. In this paper we name it **DRL Representation Consensus**.

Our work falls into the last class of work, which refines the state representation. But rather than process state information feed-forwardly (such us pooling embeddings or applying a neural network), we consider to use an auxiliary task paralleled with the main DRL task to learn semantic information for representations. Our motivation comes from the self-supervised contrastive learning in traditional Deep-Learning recommendation paradigm. However, there are three obvious problems: **(i)** In traditional recommendation paradigm, sufficient contrastive samples are derived from static datasets. But in IR scenarios, interaction history cannot provide such enough samples. **(ii)** In traditional recommendation paradigm, contrastive learning is usually used to constrain users' high-level sequence or graph representations. But directly applying it in IR will cost greatly for the large action space. **(iii)** IR models conducts online recommendation and offline training simultaneously, so contrastive learning must be conducted along with online recommendation. Whether a stable IR agent will be successfully trained in this way has not been very clear.

To tackle these problems, we propose Contrastive Representation for Interactive Recommendation (CRIR) method. The CRIR is implemented through one state representation network and our proposed Preference Ranking Contrastive Learning (PRCL). The PRCL tackles the problem (i) by fully taking advantage of users' different preference measurements towards different interacted items at every moments. The state representation network addresses problem (ii) by generating interest weights to select behavior representations which approximate the high-level user representation. This approach along with PRCL could avoid computation around whole potential action set mentioned in problem (ii). Through ranking those interests weights, a Positional Weighted InfoNCE Loss in PRCL is applied to maximize the agreements between user's preferable interests at a specific moment. Different from prior contrastive methods in DRL((Laskin, Srinivas, and Abbeel 2020; Zhang et al. 2020a)), we apply an data exploiting and agent training mechanism to solve problem (iii). In these two mechanisms, PRCL is conducted separately with main DRL task, but can achieve better effect. Extensive experiments conducted on Virtual-Taobao simulation environment and a simulator based on ml-1m dataset further verify the effectiveness of the whole proposed CRIR.

2 Related Works

Interactive Recommendation

Interactive recommendation is an online task in which agents generate recommended items and optimizes itself in the process of interacting with users. It usually models the recommendation problem as a Markov decision process and solved by RL or DRL (Lin et al. 2023; Chen et al. 2021). DRL is trained by a reward feedback evaluating its action towards the current state. But Traditional recommendation datasets are sparse and cannot give an explicit rating towards every action. So some researchers develop reward models which tracks and simulates users' behaviors from datasets or online services (Shi et al. 2019; Ie and other 2019; Rohde et al. 2018; Zhao et al. 2023). Some researchers collect some dense datasets to ease further research (Gao et al. 2022a,b).

IR has been studied from various standpoints. SlateQ (Ie et al. 2019) was proposed to decompose slate Q-value to estimate a long-term value for individual items, stating a way to recommend a page-view of items through one interaction. PGCR (Pan et al. 2019) utilized both policy gradients, time-dependent greed and actor-dropout to balance exploration and exploitation. TPGR (Chen et al. 2019a, 2023) designed a tree-structured policy gradient method to handle the large discrete action space hierarchically. Cai et al. (Cai et al. 2023b) designed two stochastic reward stabilization frameworks to replace the direct stochastic feedback with that learned by a supervised model so that to stabilize training process. In addition to general interactive recommendation, many scholars have paid attention to the practicability of IR systems. CIRS (Gao et al. 2023b) designed a causal inference based model to burst Filter Bubbles in IR. DORL (Gao et al. 2023a) made detailed analysis on Matthew Effect in IR and contribute to penalizes unbalanced exposure distribution. Dubbed RLUR (Cai et al. 2023a) focused on the user retention issue on short video IR.

Contrastive Learning in Recommender System

Contrastive Learning (CL) and Self-Supervised Learning (SSL) have brought much attentions by different research communities including CV (Chen et al. 2020; He et al. 2020a) and NLP (Gao, Yao, and Chen 2021; Zhang et al. 2020b). Some works concentrated on applying CL or SSL in DRL (Laskin, Srinivas, and Abbeel 2020; Zhang et al. 2020a) but most of which were centralized on enhancing vision encoders for RL algorithms. As far as we concerned, few works have ever tried CL for IR paradigm. We make discussions mainly on contrastive self-supervised learning in recommender system.

Applying CL in sequential recommendation models raised much attentions in recent years (Chen et al. 2022c). Xin et al. (Xin et al. 2020) used dataset labels to compute cross-entropy loss as reward to train a RL model, then used the RL model to enhance existing self-supervised sequential recommendation models in deep learning paradigm. GESU (Chen et al. 2022a) concentrated on incorporating social information to sequential recommendation models. ICL (Chen et al. 2022d) learns users' intent distributions via clustering, and then leverages the learnt intents into the user representation via their proposed contrastive approach. Graph contrastive learning also performs well on graph based recommendation tasks (Zhu et al. 2021). SGL (Wu et al. 2021) adopted a multi-task framework with contrastive SSL to improve the GCN-based collaborative filtering methods (He et al. 2020b; Wang et al. 2019). NCL (Neighborhood-enriched Contrastive Learning) (Lin et al. 2022) explicitly incorporates the potential semantic neighbors into contrastive pairs to enrich semantic information in graph. LightGCL (Cai et al. 2023c) can alleviate the problem caused by inaccurate self-supervised contrastive signals by injecting global collaboration.

Sample Efficiency in IR

As mentioned in the introduction, sample inefficiency is a tricky problem for DRL and IR (Chen et al. 2021) that still remains to be well treated. Many classical DRL methods also have many strategies to deal with sample inefficiency, e.g. PPO (Schulman et al. 2017), SAC (Haarnoja et al. 2018b,a), CRR (Wang et al. 2020), DDPG (Lillicrap et al. 2016). However, those naive DRL methods are not enough to treat with IR scenarios. Many works have broadened new horizons to make interactive recommendation more reliable. DRR (Liu et al. 2020) proposed some basic state representation method and a generative recommendation paradigm utilizing DDPG. NICF (Zou et al. 2020) designed an exploration policy with multi-channel transformer to capture users' shifting interest in cold-start settings. KGRL (Zhou et al. 2020) utilized knowledge graph to enhanced semantic information in reinforcement learning. Xi et al. (Xi et al. 2023) used transformer as state representation network and CRR as backbone RL framework along with pre-trained embeddings to make recommendation. LSER (Chen et al. 2022b) applied Locality-Sensitive Hashing algorithm in experience replay procedure to sample most valued training batches. DACIR (Wu et al. 2022) aligned embeddings from different domain into a shared latent space to fertilize embedding information for cross-domain interactive tasks.

Although IR has gained significant attention recently, research on its sample efficiency remains neither sufficient nor systematic. Some recent works are noteworthy, but they address different problems or are applied in very different contexts (such as TPGR, DORL, KGRL, LSER, DACIR, etc.). Consequently, our options for baseline are limited. So we choose SAC, CRR, PPO, DRR, and NICF as baselines.

3 Contrastive Representation

Framework Preliminaries

CRIR uses the auxiliary, paralleled task PRCL to get better representations for main DRL task. In this paper we name this training mechanism as **Auxiliary Mechanism.** As shown in Figure 1, the proposed Contrastive Representation is composed of a State Representation Network and the PRCL method. They cooperate to acquire high level representations through the connection of *Interest Weight*. The *Interest Weight* is utilized to formulate *State Representation* for RL, and also indicate the importance of the interacted items in PRCL at every specific moment. The replay buffer is a general components in off-policy DRL (Lillicrap et al. 2016). Here it stores historical interaction transitions. It will sample a batch of transitions while training the agent and conducting PRCL. Each transition contains one users' interaction history and other profiles at one past moment. In our implementation, we use DDPG (Lillicrap et al. 2016) along with Priority Experience Replay mechanism (PER) (Schaul et al. 2015) as our DRL backbone for its effectiveness and stability.

State Representation Network

Some works have already employ attention mechanism or transformer (Vaswani et al. 2017) to model state representation in IR (Liu et al. 2020; Gao et al. 2023b; Xi et al. 2023). However, what we need is the explicit degree of emphasis to different behaviors of the user at a specific timestamp. We discover that the weighted sum attention mechanism in Deep Interest Network (Zhou et al. 2018) naturally fits this paradigm. Its effectiveness is also validated by various online recommendation services. So it is determined as part of the state representation network to model the state information and generate preference scores of interacted items.

As shown in Figure 2, features and behavior histories of the current user are fed into their respective embedding layers. Behavior history contains not only item features but also feedback given by the user at each moment. Then weights for each single behavior will be computed through each activation unit, whose specific structure is shown in Figure 2. The settings for activation unit and Dice activation function follow Deep Interest Network (Zhou et al. 2018).

Average information should be preserved to retain basic state information and stabilize convergence. This idea is proved to be effective in DRR (Liu et al. 2020). So we employ the average pooling in parallel with the weighted sum attention module. The final state representation of user u at timestamp t is formulated as:

$$s_{u,t} = (\frac{1}{t}\sum_{\tau=1}^{t} u_t \otimes h_\tau) \oplus (\sum_{\tau=1}^{t} \Lambda(u_t, h_\tau) \cdot h_\tau), \quad (1)$$

where $\Lambda(\cdot,\cdot) \in \mathbb{R}$ is the activation unit, with representations of user $u_t \in \mathbb{R}^{D_R}$ and behaviors $h_\tau \in \mathbb{R}^{D_R}$ as input, D_R is the representation dimension, $\otimes$ and $\oplus$ stands for the outer product and concatenation separately.

Preference Ranking Contrastive Learning

This section will specifically states our proposed PRCL. We will first give a brief introduction to the problem definition and then specifically introduce the procedure of PRCL, including Data Augmentation and Positional Weighted InfoNCE Loss. Figure 1 could vividly show the process.

IR Objective The optimization objective of the whole IR process could be formulated as maximizing $J_{\omega,\theta}$:

$$J_{\omega,\theta} = \sum_{u=1}^{|\mathcal{U}|}\sum_{t=1}^{T_u} \mathbb{E}_{a_{t;\theta}}[r(s_{u,t;\omega}, a_{u,t;\theta})], \quad (2)$$

where ω, θ are the parameter sets for state representation and DRL components, separately. $\mathcal{U}$ is the user set, T_u is the interaction length for user u, $s_{t;\omega} \in \mathbb{R}^{D_S}$ is the representation for user state s at timestamp t with parameter set ω. $r(s_t, a_t)$ is the reward returned from environment while taking action a_t at state s_t. The action $a_{t;\theta} \in \mathbb{R}^D$ is actually the representation of the chosen item to be recommended. This maximization goal could be transformed to the goal particularly for PRCL task around user u:

$$J_{u;\omega} = \sum_{t=1}^{T_u}\sum_{k}^{|\mathcal{A}_{s_t}|} \ln P(s_{t;\omega}, a_{t,k}), \quad (3)$$

where $P(s_{t;\omega}, a_{t,k})$ is the joint probability for the agent taking action $a_{t,k}$ at state s_t, $\mathcal{A}_{s_t}$ is the potential action set for state $s_{t;\omega}$. Our PRCL is concentrated on optimizing Eq (3).

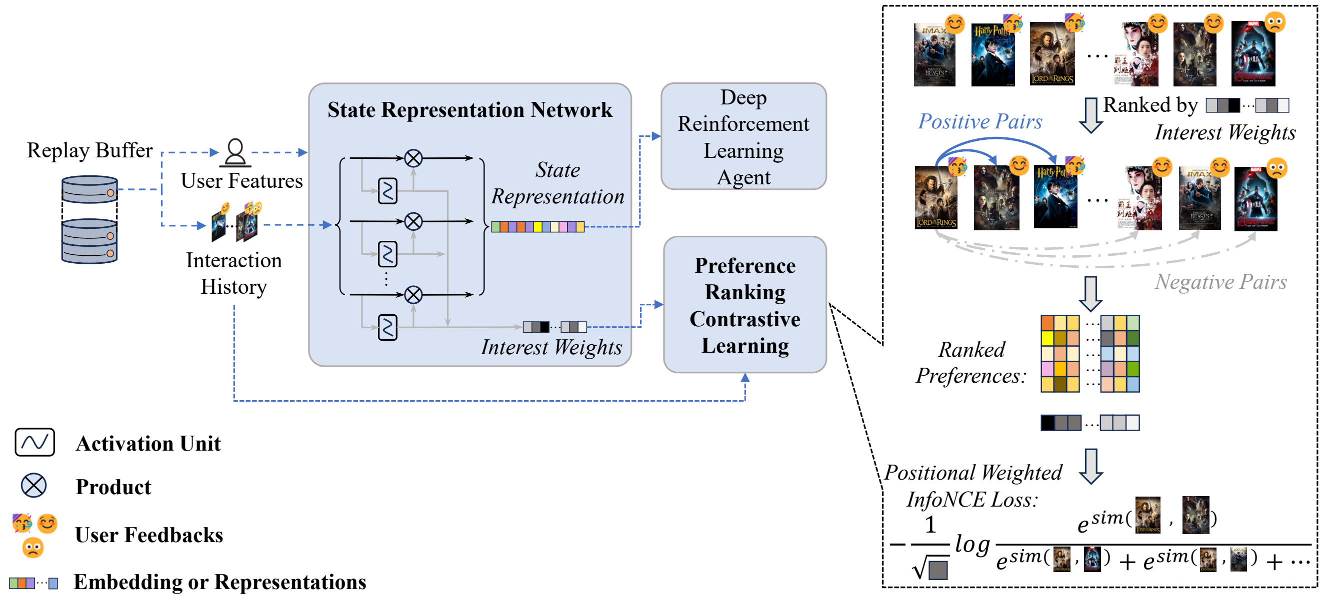


Figure 1: Overview of Contrastive Representation for Interactive Recommendation.

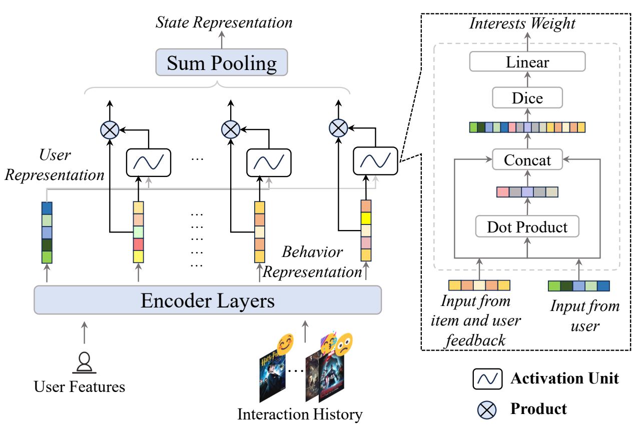


Figure 2: Weighted sum part of the state representation network ($\sum_{\tau=1}^{t} \Lambda(u_t, h_\tau) \cdot h_\tau$ in equation Eq (1)).

Data Augmentation

(i) Sampling: The replay buffer samples history interactions to train the DRL agent. We should also sample data for PRCL. Considering that PRCL is conducted at different stage with the main DRL task, we design an data sampling mechanism, which can achieve both two optimization goals simultaneously. In our implementation, we use two batches of data to conduct contrastive learning. One is sampled in totally random from the replay buffer, and another is the data for training the DRL networks in the next round, using the PER sampling strategy. This mechanism ensures that every transition undergoing reinforcement learning also experiences contrastive learning at least once. In this paper we name this data exploiting mechanism as **Mixed Mechanism**. Therefore, the goal of DRL and PRCL could be achieved together although they are conducted separately. Our experiments will study the Mixed Mechanism specifically.

(ii) Weighting: As shown in Figure 2, the state representation network can either model the state information or generate interest weights for different behaviors. One single behavior with larger weight value means that the current user is predicted to pay more attention to the item in this behavior. These weights plays critical roles at the following **(iii) Ranking** step and Positional Weight InfoNCE Loss.

(iii) Ranking: Every interaction in the sampled batch is assigned with an interest weight as mentioned in **(ii) Weighting**. Suppose the length for an interaction history is n with max sequence length M ($n \leq M$). The ranked list of the behavior representation sequence is formed as $[h_1, h_t, \cdots, h_n]$ with higher interest weight ranking ahead.

Then positive and negative pairs for contrastive learning should be generated. In every interaction, the attention scores ranking the second to the the $\lfloor n/2 \rfloor$-th will be treated as candidate positive items. Randomly choose $k \in \{2, \cdots, \lfloor n/2 \rfloor\}$, then (h_1, h_k) is treated as the positive pair. Every tuple like (h_1, h_t) where $t \in \{\lfloor n/2 \rfloor + 1, \cdots, n\}$ is treated as negative pairs for this interaction. Finally, we get one positive pair and $\lceil n/2 \rceil$ negative pairs for each transition in the training batch to conduct PRCL.

Positional Weight InfoNCE Loss Since it is reasonable for the agent to make action according to the current state, it's reasonable for this distribution of action $a_{t,k}$ in Eq (3) to be written as Gaussian-distribution-like loss around state s_t:

$$\mathcal{L}_u = -\sum_{t=1}^{T_u} \sum_{k}^{|\mathcal{A}_{s_t}|} \log \frac{\exp(-a_{t,k}^{\mathrm{T}} W s_t)}{\sum_{j=1}^{|\mathcal{A}_{s_t}|} \exp(-a_{t,j}^{\mathrm{T}} W s_t)}, \quad (4)$$

where $W \in \mathbb{R}^{D \times D_S}$. In order to simplify the computational complexity, we utilize representative behavioral representation to approximate the state s_t and the sum-up operation around potential action set $\mathcal{A}_{s_t}$. The optimization goal for user u at timestamp t could be formulated as:

$$\mathcal{L}_u(t) = -\log \frac{\exp(-h_k^{\mathrm{T}} \cdot h^*)}{\sum_{n=1}^{|\mathcal{N}_{s_t}|} \exp(-h_n^{\mathrm{T}} \cdot h^*)}, \tag{5}$$

where $h^* = h_i,\ i = \underset{i \le t}{\operatorname{argmax}}\ w_i,\ t \le T_u$ is the optimal behavioral representation, k is the chosen behavior index from positive set mentioned in **(iii) Ranking**, $\mathcal{N}_{s_t}$ is the negative behavior set at current state s_t, also mentioned in **(iii) Ranking**. $\mathcal{N}_{s_t}$ could be seen as negative sampling, utilized to alternate computation on the whole potential action set.

Considering that different h_k should have different similarity value with h^*, we decide to use a coefficient to model this discrimination of different contrastive pairs. Obviously the ranking position can measure the importance of the contrastive pair in training the representation. **So we use $\mathbf{1/\sqrt{R_u(h_k)}}$ to smooth the discrimination**, where $R_u(h_k)$ is the ranking position for item h_k of user u mentioned at **(iii) Ranking**. The proposed Positional Weighted InfoNCE Loss is formulated as:

$$\mathcal{L}_u(t) = -\frac{1}{\sqrt{R_u(h_k)}} \log \frac{\exp(-h_k^{\mathrm{T}} \cdot h^*)}{\sum_{n}^{\mathcal{N}_{s_t}} \exp(-h_n^{\mathrm{T}} \cdot h^*)}. \tag{6}$$

4 Experiments

In experiment section we want to investigate the following research questions.

- **(RQ1)** How does CRIR perform with other IR methods aiming at improving sample efficiency?
- **(RQ2)** What contributions dose each PRCL components make in the whole system?
- **(RQ3)** Does the sampling and training mechanism contribute greatly to the training performance?

Experimental Setup

Recommendation Environment Traditional recommendation datasets are too sparse to evaluate the interactive recommender systems (Gao et al. 2023a). Because instant feedback is demanded at every timestamp in interactive settings. Dataset can hardly reflect this. So we use **Virtual-Taobao** (Shi et al. 2019) and a dataset-oriented **simulator based on ML-1M** [1] to evaluated CRIR and baseline methods. These simulators will generate a reward signal towards every recommendation reflecting the performance, which satisfy our problem settings. Specifically, we add some dynamic features, like shifting interest, to the ML-1M-based simulator. This is intended for verifying whether experimental methods could perfectly catch dynamic information in recommendation environment.

To fully investigate the sample efficiency of each model, we conduct our experiment **in totally cold-start settings**, which means all representation parameters are randomly initialized. The model with superior sample efficiency can quickly learn features of users and items from scratch.

Evaluation Metrics We use two widely used metrics in IR: **Cumulative Reward** $\sum_t r_t$ in an episode and **Click Through Rate (CTR)** as our evaluation metric, following previous IR works (Gao et al. 2023b; Chen et al. 2022b; Gao et al. 2023a). Here CTR is denoted as the proportion of positive rewards among all rewards in an episode. Positive reward is denoted as those rewards greater than 0 in both of the two simulation environment.

Sample efficiency is measured through the training effect within the same quantity of data (Mai, Mani, and Paull 2022). So we use line chart rather than static table to fully display experimental result at every episode.

There are two reasons why IR cannot be evaluated by list-wise accuracy indicators such as NDCG@K, HR@K. One reason is that precision-based metrics cannot reflect the performance of decision tasks (Gao et al. 2023a). Another reason is that IR usually applies generative recommendation method rather than scoring-and-ranking method.

Baselines The reason why we chose these baselines is in Section 2.

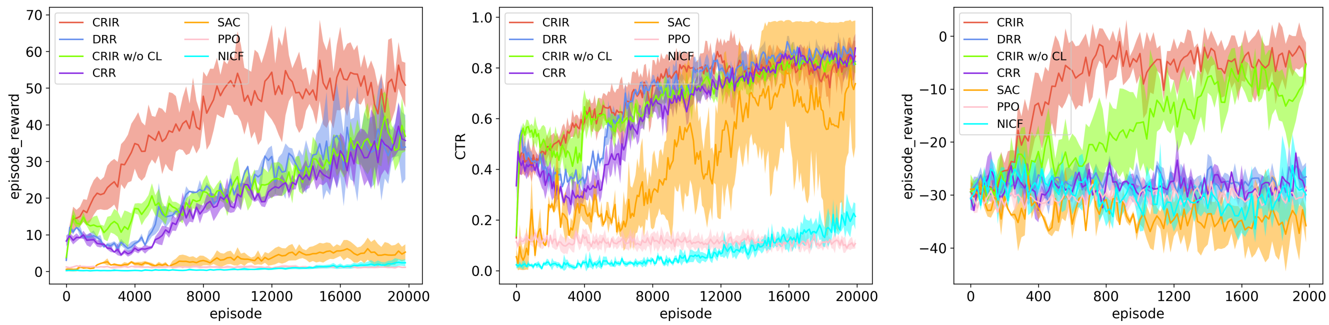
- **SAC**, named Soft Actor Critic, utilized action distribution construct an entropy to constrain action space.
- **CRR**, named Critic Regularized Regression, is a model-free RL method that improve sample efficiency by regularizing weights for policy learning.
- **PPO**, named Proximal Policy Optimization, optimizes a surrogate objective function with gradient ascent while limiting the policy update size to ensure stability.
- **DRR** explored some feasible state representations and investigated a basic generative paradigm applying DDPG for IR.
- **NICF**, named Neural Interactive Collaborative Filtering, utilize Q-learning and multi-channel transformer to enhance the exploration policy.
- **CRIR w/o CL (Ablation Study)** is our CRIR method without our PRCL approach. It only enhanced the state representation network . This experiment is conducted to verify the contribution of PRCL.

Similar to the DRR, CRIR also use DDPG as implementation backbone and utilize a generative recommendation paradigm. So through the comparison between DRR and CRIR w/o CL, the contribution of the designed state representation network could be verified.

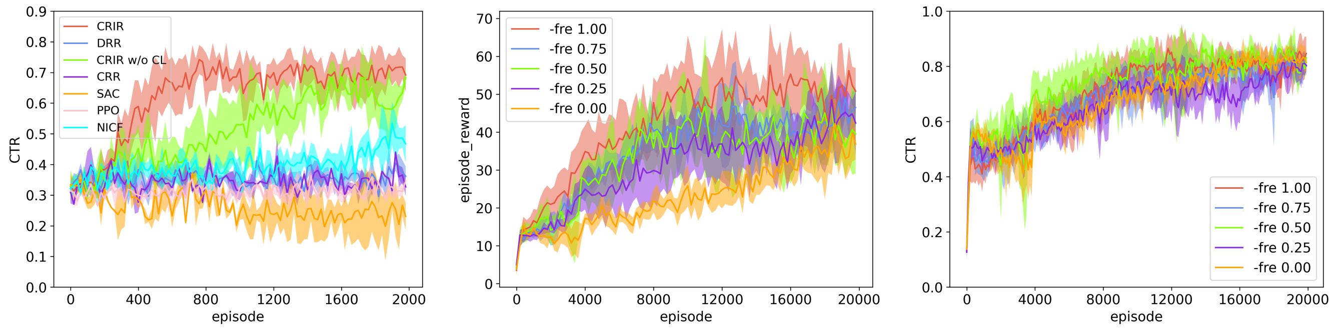
Overall Performance and Ablation Study (RQ1)

We first make observations on Virtual-Taobao environment. Figure 3 (a) shows the cumulative reward metric. Our CRIR approach outperforms the others in the Virtual-Taobao environment under cold-start settings. It takes the lead in finding a good recommendation policy at around 8000-th episode, while the others have not reached this level within 20000 episodes. The ablation study between whole CRIR and CRIR w/o CL, as well as that between CRIR w/o CL and DRR, demonstrate the contribution of PRCL and the state

[1] https://grouplens.org/datasets/movielens/1m/



(a) Episode reward for Virtual-Taobao (b) CTR for Virtual-Taobao (c) Episode reward for ML-1M



(d) CTR for ML-1M (e) Episode rewards for PRCL frequency study (f) CTR for PRCL frequency study

Figure 3: Performance for the proposed, ablation and baseline methods in cold-start setting. Each curve in the graph is repeated for 5 times and 95% confidence intervals are depicted. (a)-(d) are the results for RQ1, (e) and (f) are for RQ2-1.

representation network, separately. But the improvement is slight by only use the representation network. The CRIR w/o CL method performs better than DRR, SAC, CRR, PPO and NICF in early stage, but fails to keep up with CRIR, and gradually declines to the same with the others. Its representation structure helps capture the user's interest initially but fails to make further progress in subsequent episodes. As CTR metric depicted in Figure 3 (b) shows, most of the models finally rise up to around 0.8. Note that the reward signals for Virtual-Taobao are greater or equals to 0, which makes high CTR scores easy to achieve. So through the comparison between cumulative reward and CTR, we can know that although most baselines finally reach the same level with CRIR at CTR metric, they get less high-reward actions than CRIR. Baseline methods expect CRIR w/o CL still suffer from sampling inefficiency before 6000-th episode. SAC can scarcely rise but sometimes succeed in CTR metric. PPO fails to learning a correct policy in cold-start settings. The reason for this is that on-policy methods like PPO could hardly filter unimportant or blured transitions from cold-start representations. These on-policy methods usually require well pre-trained representations. NICF is designed specifically for discrete action space initially, it seems not compatible with continuous environment like Virtual-Taobao.

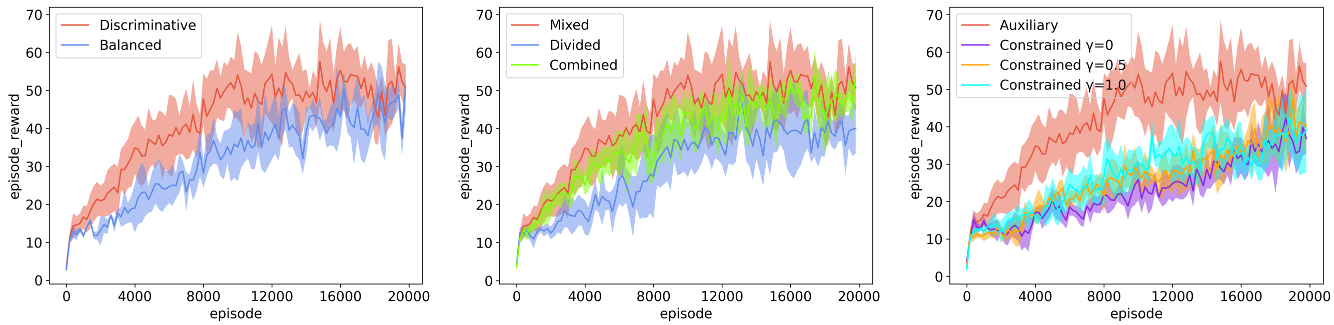
Then we make observations on ML-1M-based environment. Figure 3 (c) shows cumulative reward metric, CRIR outperforms the best on ML-1M oriented simulator. CRIR w/o CL converges slower than CRIR but outperforms all the other methods. Methods except PRCL , CRIR w/o CL and NICF fails within 2000 episodes in this simulator. One reason is that the dynamic features of user interest change quickly in the simulator. The other reason lies in the cold-start setting. These methods do not have enough sample efficiency to find valid policy in such settings. The CTR metric depicted in Figure 3 (d) seems very consistent with the cumulative reward metric shown in Figure 3 (c). The reason for this phenomenon is that the simulator returns rewards ranged from -1 to 1, with positive value roughly equal with negative values.

In summary, experiments conducted in two environments demonstrate the effectiveness of CRIR in improving sample efficiency. The comparison between CRIR w/o CL, DRR and CRIR confirms the utility of CRIR' state representation network and PRCL method.

Contribution Quantitative Study (RQ2)

We conduct two quantitative contribution studies on two key factors of CRIR to study their detailed contributions. The first experiment studies quantitative research on different frequency of the PRCL. The frequency of PRCL is denoted as the ratio of the times PRCL conducted in that of the RL. The second experiment studies the significance of the discriminative coefficients $1/\sqrt{R_u(h_k)}$ in equation Eq. (6). We make these two experiments on Virtual-Taobao.

For the first experiment (**RQ2-1**), we set the PRCL frequencies in $\{0, 0.25, 0.5, 0.75, 1.0\}$. Episode reward and



(a) (RQ2-2) Different coefficient strategy (b) (RQ3-1) Different sampling mechanism (c) (RQ3-2) Different training mechanism

Figure 4: Study on coefficient strategy, data sampling and agent training mechanism of PRCL. Each curve is repeated for 5 times and 95% confidence intervals are depicted.

CTR are shown in Figure 3 (e) and (f) separately. The sample efficiency is boosted with the increase of PRCL frequency. The increment of performance seems not linear. The difference between 0.25 and 0.5 is much larger than that between 0.5 and 0.75. It states that PRCL can effectively improve sample efficiency. But the increment has a limit while increasing the frequency. And PRCL is more capable of optimizing hard metrics like episode reward, than easy metrics like CTR.

For the second experiment (**RQ2-2**), we use balanced coefficients w to replace the $1/\sqrt{R_u(h_k)}$ in equation Eq. (6) as baseline method. To guarantee the same average intensity for contrastive learning, we set all the coefficients to $w = (1/\lfloor T/2 \rfloor) \sum_{i=2}^{\lfloor T/2 \rfloor} (1/\sqrt{i}) \approx 0.3183$ where $T = 50$ is the max sequence length for state representation. The result can be seen in Figure 4(a). As the result shown, the discriminative coefficient strategy performs better than the balanced one. The balanced strategy performs approximately the same with the PRCL with learning frequency of 0.25 in Figure 3(e). This demonstrates the utility of the proposed discriminative coefficients in PRCL.

Sampling and Training Mechanism Study (RQ3)

We will verify our proposed data sampling and training mechanism mentioned in Section 3 (**RQ3-1**). The proposed data sampling mechanism utilizes two batches of interaction data for PRCL — one is the batch planed to train DRL immediately (sampled by PER strategy) while the another is randomly sampled from the replay buffer. This sampling strategy is named as **Mixed Mechanism**. Accordingly, we may consider other two mechanisms — totally sampling randomly from the buffer, or just using the data planed for DRL training. We name them **Divided Mechanism** and **Combined Mechanism** separately.

Considering that our PRCL task is conducted independently with the DRL, we also study the dependent way to conduct PRCL (**RQ3-2**). In DRL task, our state representation network is updated by value function (critic network) in DRL. So it means that the Positional Weighted InfoNCE Loss is added to the loss of the value function as a constraint. In this way the loss of value function is formulated as:

$$\mathcal{L} = \frac{1}{2}\delta^2 + \gamma * \mathcal{L}_{PRCL} \tag{7}$$

where δ is the TD-error in DRL, γ is hyper-parameter that controls the strength of PRCL, and $\mathcal{L}_{PRCL}$ is defined in Eq (6). We name this training mechanism as **Constrained Mechanism**. We choose $\gamma \in \{0, 0.5, 1.0\}$. Conversely, we name CRIR's training strategy as **Auxiliary Mechanism**.

As shown in Figure 4(b), the Mixed Mechanism performs the best among all data sampling strategies. The comparison between Mixed, Divided and Combined Mechanism demonstrates the effectiveness of our Mixed Mechanism. It shows that representation will be learnt better by utilizing DRL training samples along with some extra samples. As shown in Figure 4(c), the γ value have little effect on the performance. PRCL seems to have no improvement in Constrained Mechanism. This demonstrates the effectiveness of our Auxiliary training strategy.

5 Conclusion

This paper states that sample inefficiency is a tricky problem that hinders the development of IR. Inspired by contrastive learning in traditional recommendation paradigm, we propose Contrastive Representation for Interactive Recommendation (CRIR), which contains a state representation network and Preference Ranking Contrastive Learning (PRCL). These two methods could help the agent learns better representations. Then the sample efficiency is improved according to the DRL Representation Consensus. Different from precious works, we apply an auxiliary contrastive learning task in parallel with the main DRL task. We also adopt an data sampling strategy to ensure the different optimization goals will not be conflicted. Extensive experiments have verified the effectiveness of the the proposed CRIR.

Acknowledgments

Zhiyong Feng is the corresponding author. This work was supported by the National Natural Science Foundation of China (NSFC) (Grant Numbers 62372323, 62422210, 62276187).

References

Cai, Q.; Liu, S.; Wang, X.; Zuo, T.; Xie, W.; Yang, B.; Zheng, D.; Jiang, P.; and Gai, K. 2023a. Reinforcing User Retention in a Billion Scale Short Video Recommender System. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, 421–426.

Cai, T.; Bao, S.; Jiang, J.; Zhou, S.; Zhang, W.; Gu, L.; Gu, J.; and Zhang, G. 2023b. Model-Free Reinforcement Learning with Stochastic Reward Stabilization for Recommender Systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, 2179–2183.

Cai, X.; Huang, C.; Xia, L.; and Ren, X. 2023c. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *The Eleventh International Conference on Learning Representations*.

Chen, H.; Dai, X.; Cai, H.; Zhang, W.; Wang, X.; Tang, R.; Zhang, Y.; and Yu, Y. 2019a. Large-scale interactive recommendation with tree-structured policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 3312–3320.

Chen, H.; Feng, Z.; Chen, S.; Xue, X.; Wu, H.; Sun, Y.; Xu, Y.; and Han, G. 2022a. Capturing Users' Fresh Interests via Evolving Session-Based Social Recommendation. In *2022 IEEE International Conference on Web Services (ICWS)*, 182–187.

Chen, H.; Zhu, C.; Tang, R.; Zhang, W.; He, X.; and Yu, Y. 2023. Large-Scale Interactive Recommendation With Tree-Structured Reinforcement Learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(4): 4018–4032.

Chen, M.; Beutel, A.; Covington, P.; Jain, S.; Belletti, F.; and Chi, E. H. 2019b. Top-K Off-Policy Correction for a REINFORCE Recommender System. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, 456–464.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20.

Chen, X.; Yao, L.; McAuley, J.; Guan, W.; Chang, X.; and Wang, X. 2022b. Locality-Sensitive 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*, SIGIR '22, 1316–1325.

Chen, X.; et al. 2021. A Survey of Deep Reinforcement Learning in Recommender Systems: A Systematic Review and Future Directions. *ArXiv*, abs/2109.03540.

Chen, Y.; Liu, Z.; Li, J.; McAuley, J.; and Xiong, C. 2022c. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, 2172–2182.

Chen, Y.; Liu, Z.; Li, J.; McAuley, J.; and Xiong, C. 2022d. Intent Contrastive Learning for Sequential Recommendation. In *WWW '22*, 2172–2182.

Gao, C.; Huang, K.; Chen, J.; Zhang, Y.; Li, B.; Jiang, P.; Wang, S.; Zhang, Z.; and He, X. 2023a. Alleviating Matthew Effect of Offline Reinforcement Learning in Interactive Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, 238–248.

Gao, C.; Li, S.; Lei, W.; Chen, J.; Li, B.; Jiang, P.; He, X.; Mao, J.; and Chua, T.-S. 2022a. KuaiRec: A Fully-observed Dataset and Insights for Evaluating Recommender Systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, 540–550.

Gao, C.; Li, S.; Zhang, Y.; Chen, J.; Li, B.; Lei, W.; Jiang, P.; and He, X. 2022b. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, 3953–3957.

Gao, C.; Wang, S.; Li, S.; Chen, J.; He, X.; Lei, W.; Li, B.; Zhang, Y.; and Jiang, P. 2023b. CIRS: Bursting filter bubbles by counterfactual interactive recommender system. *ACM Transactions on Information Systems*, 42(1): 1–27.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6894–6910.

Haarnoja; et al. 2018a. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018b. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. PMLR.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020a. Momentum Contrast for Unsupervised Visual Representation Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9726–9735.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020b. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, 639–648.

Ie, E.; Jain, V.; Wang, J.; Narvekar, S.; Agarwal, R.; Wu, R.; Cheng, H.-T.; Chandra, T.; and Boutilier, C. 2019. SLATEQ: A Tractable Decomposition for Reinforcement Learning with Recommendation Sets. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, 2592–2599.

Ie, E.; and other. 2019. Recsim: A configurable simulation platform for recommender systems. *arXiv preprint arXiv:1909.04847*.

Kaiser, L.; et al. 2024. Model-Based Reinforcement Learning for Atari. arXiv:1903.00374.

Lake, B. M.; Ullman, T. D.; Tenenbaum, J. B.; and Gershman, S. J. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40: e253.

Laskin, M.; Srinivas, A.; and Abbeel, P. 2020. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR*.

Lin, Y.; Liu, Y.; Lin, F.; Zou, L.; Wu, P.; Zeng, W.; Chen, H.; and Miao, C. 2023. A Survey on Reinforcement Learning for Recommender Systems. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21.

Lin, Z.; Tian, C.; Hou, Y.; and Zhao, W. X. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022*, 2320–2329.

Liu, F.; Tang, R.; Li, X.; Zhang, W.; Ye, Y.; Chen, H.; Guo, H.; Zhang, Y.; and He, X. 2020. State representation modeling for deep reinforcement learning based recommendation. *Knowledge-Based Systems*, 205: 106170.

Mai, V.; Mani, K.; and Paull, L. 2022. Sample Efficient Deep Reinforcement Learning via Uncertainty Estimation. In *International Conference on Learning Representations*.

OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Pan, F.; Cai, Q.; Tang, P.; Zhuang, F.; and He, Q. 2019. Policy Gradients for Contextual Recommendations. In *The World Wide Web Conference*, WWW '19, 1421–1431.

Rohde, D.; et al. 2018. RecoGym: A Reinforcement Learning Environment for the problem of Product Recommendation in Online Advertising. *arXiv preprint arXiv:1808.00720*.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2015. Prioritized experience replay. *International Conference on Learning Representations*.

Schulman; et al. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shi, J.-C.; Yu, Y.; Da, Q.; Chen, S.-Y.; and Zeng, A.-X. 2019. Virtual-Taobao: Virtualizing Real-World Online Retail Environment for Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 4902–4909.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, 165–174.

Wang, Z.; Novikov, A.; Zolna, K.; Merel, J. S.; Springenberg, J. T.; Reed, S. E.; Shahriari, B.; Siegel, N.; Gulcehre, C.; Heess, N.; et al. 2020. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33: 7768–7778.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 726–735.

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

Xi, X.; Zhao, Y.; Liu, Q.; Ouyang, L.; and Wu, Y. 2023. Integrating Offline Reinforcement Learning with Transformers for Sequential Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*.

Xin, X.; Karatzoglou, A.; Arapakis, I.; and Jose, J. M. 2020. Self-Supervised Reinforcement Learning for Recommender Systems. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Yu, T.; Shen, Y.; and Jin, H. 2019. A Visual Dialog Augmented Interactive Recommender System. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, 157–165.

Yu, Y. 2018. Towards Sample Efficient Reinforcement Learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 5739–5743.

Zhang, A.; et al. 2020a. Learning Invariant Representations for Reinforcement Learning without Reconstruction. *ArXiv*, abs/2006.10742.

Zhang, Y.; He, R.; Liu, Z.; Lim, K. H.; and Bing, L. 2020b. An Unsupervised Sentence Embedding Method by Mutual Information Maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1601–1610.

Zhao, K.; Liu, S.; Cai, Q.; Zhao, X.; Liu, Z.; Zheng, D.; Jiang, P.; and Gai, K. 2023. KuaiSim: A Comprehensive Simulator for Recommender Systems. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1059–1068.

Zhou, S.; Dai, X.; Chen, H.; Zhang, W.; Ren, K.; Tang, R.; He, X.; and Yu, Y. 2020. Interactive Recommender System via Knowledge Graph-Enhanced Reinforcement Learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, 179–188.

Zhu, Y.; et al. 2021. An Empirical Study of Graph Contrastive Learning. *ArXiv*, abs/2109.01116.

Zou, L.; Xia, L.; Gu, Y.; Zhao, X.; Liu, W.; Huang, J. X.; and Yin, D. 2020. Neural Interactive Collaborative Filtering. In *SIGIR '20*, 749–758.