中国科学技术大学
University of Science and Technology of China

快手
KUAISHOU TECHNOLOGY

CIRS: Bursting Filter Bubbles by Counterfactual Interactive Recommender System

Chongming Gao[1] , Wenqiang Lei[2] , Jiawei Chen[1] , Shiqi Wang[3] , Xiangnan He[1,] , Shijun Li[1] , Biao Li[4] , Yuan Zhang[4] , Peng Jiang[4]*

[1]University of Science and Technology of China; [2]Sichuan University, China; [3]Chongqing University, China; [4]Kuaishou Technology Co., Ltd

https://chongminggao.me | chongming.gao@gmail.com

❖ Outline

1. **Background and Motivation.**
   - Filter Bubbles in Recommendation
   - Why Do we Choose the Interactive Recommendation?
   - Empirical Study of User Satisfaction in Filter Bubbles
   - Motivation of the idea
2. **Related Works and Existing Problems**
3. **Proposed Method: CIRS**
4. **Experiments**

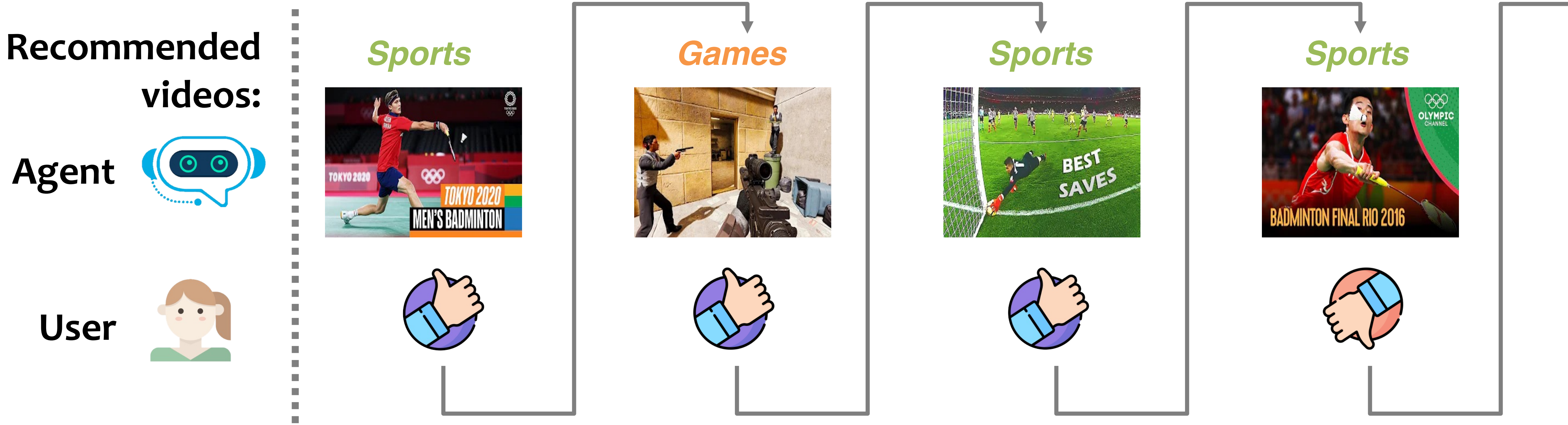
1.1 Filter Bubbles in Recommendation

- Filter bubble
  - The phenomenon that recommender **emphasizes only a small set of items in the feedback loop** of the interaction process
  - Similar concepts: echo chamber, information cocoon

Filter bubbles in the recommendation-feedback loop

1.2 Why Interactive Recommender System (IRS)?

- Because IRS is **the general form** of real-world recommenders *(static recommender is only a special/simplified case of IRS).*
- Because IRS **provides an environment to evaluate** the effect of filter bubbles.



An interaction trajectory in Kuaishou, a video viewing App

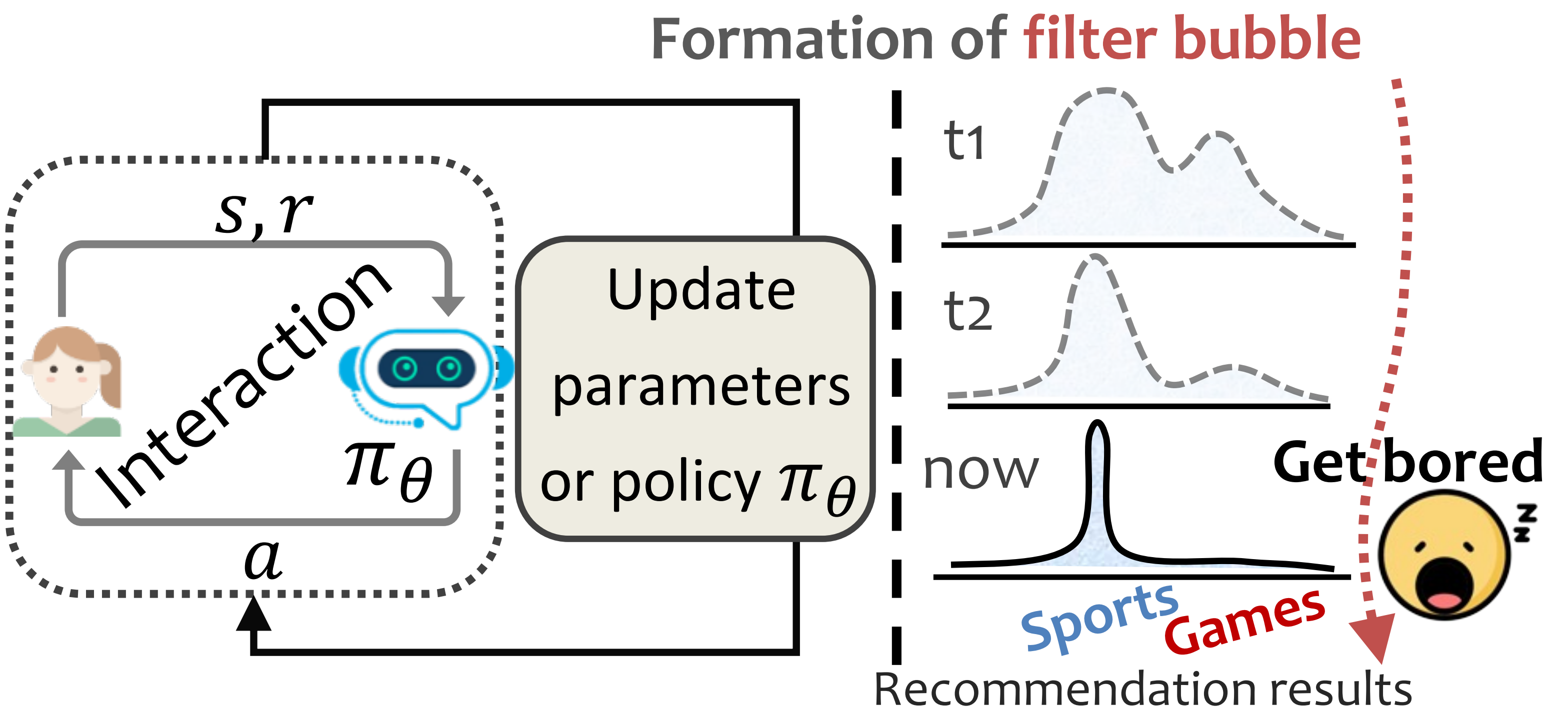
1.2 Why Interactive Recommender System (IRS)?

- Because IRS is **the general form** of real-world recommenders *(static recommender is only a special/simplified case of IRS).*
- Because IRS **provides an environment to evaluate** the effect of filter bubbles.

s: the **state** representing the context of the interaction.

r: the **reward** representing user satisfaction.

a: an **action**, e.g. a recommended item



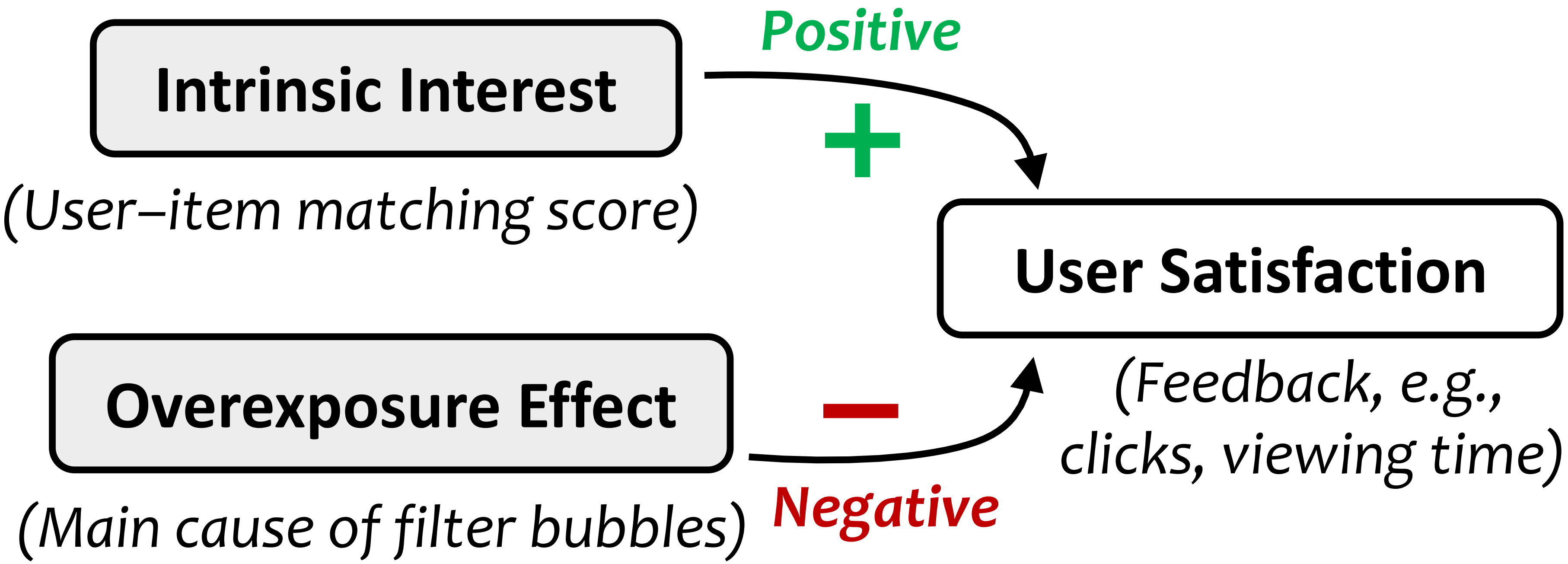
(a) Interactive recommendation (b) Filter bubble

The general framework of interactive recommendation

1.3 User Satisfaction in Filter Bubbles

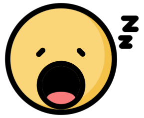
Get bored

- **Assumption**: Users may **feel bored and give negative feedback** in such a repeated and monotonous recommendation stream.

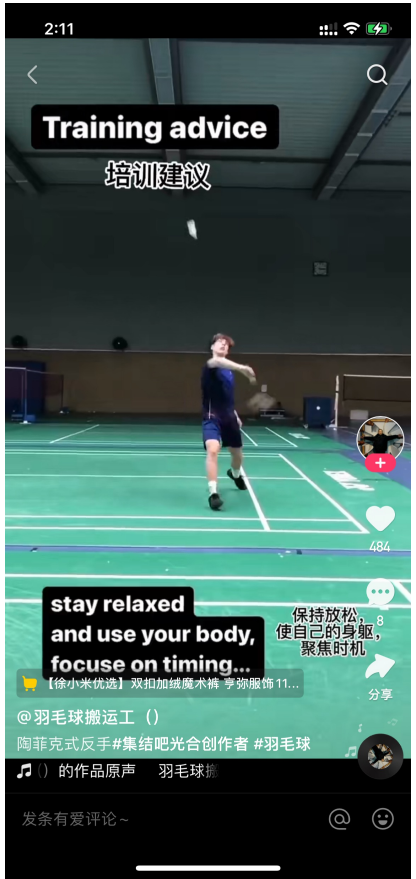


1.3 User Satisfaction in Filter Bubbles

Get bored



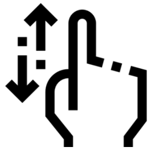
- Assumption: Users may **feel bored and give negative feedback** in such a repeated and monotonous recommendation stream.
- Empirical studies conducted on Kuaishou App.

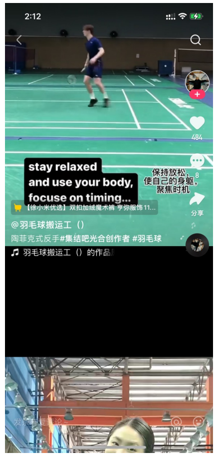


The recommended video stream in Kuaishou App

Two important user behaviors reflecting satisfaction

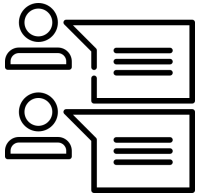
① Keeping watching until quitting or scrolling to the next one





*Metric: **Watching ratio***
(watching time / video video time duration)

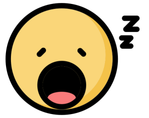
② Hitting and staying in the comments section



*Metric: **Time staying comments section***

1.3 User Satisfaction in Filter Bubbles

Get bored



- **Assumption**: Users may **feel bored and give negative feedback** in such a repeated and monotonous recommendation stream.

Empirical study conducted on:

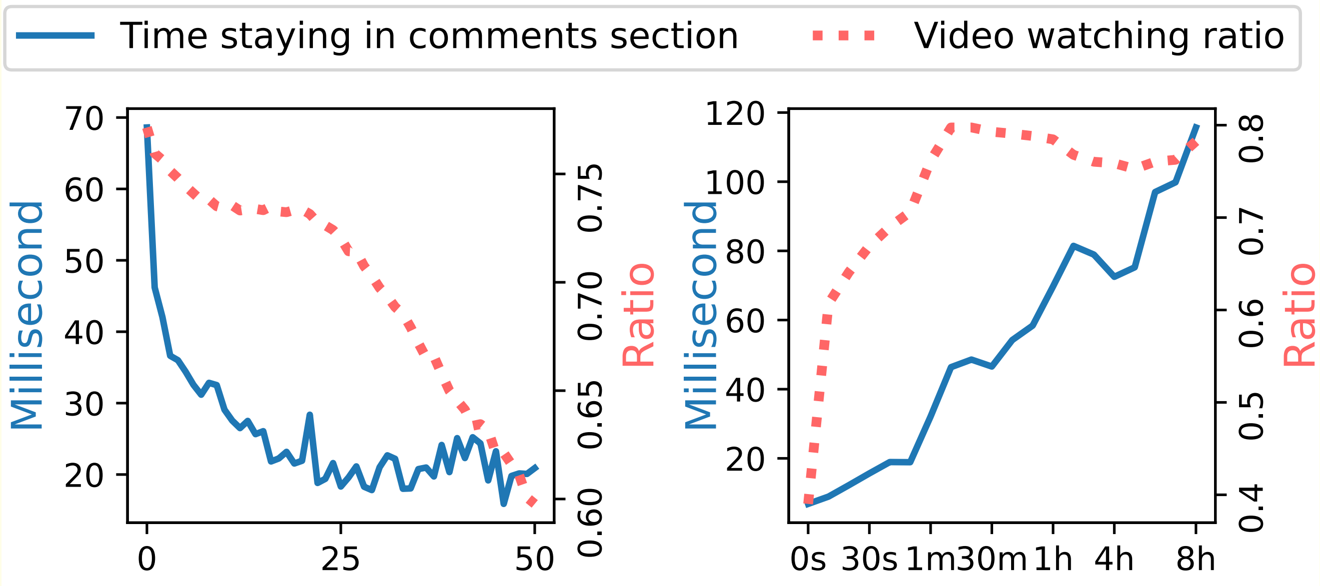
7176 users

3,110,886 videos

31 category tags for videos

34,215,294 views in total

August 5th – August 11st, 2020



(a) Number of similar videos watched in one hour

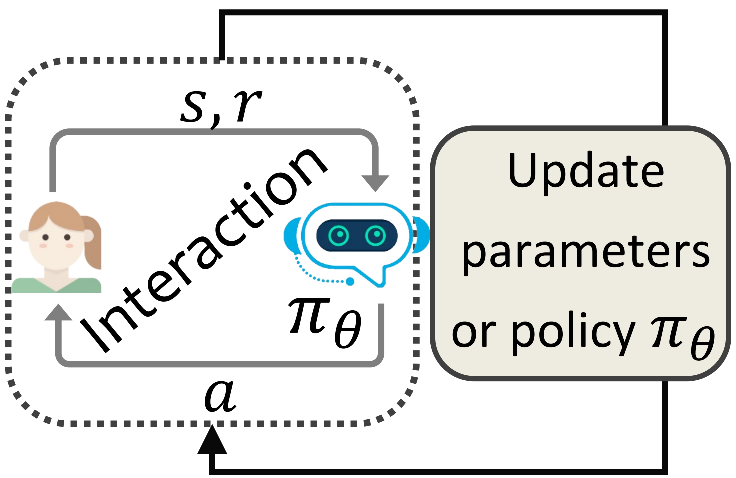
(b) Time interval between now and last time seeing a similar video

Observation 1: *User satisfaction towards a recommended item drops when the system increases the number of similar items that have the same categories with this item in recent recommendations.*

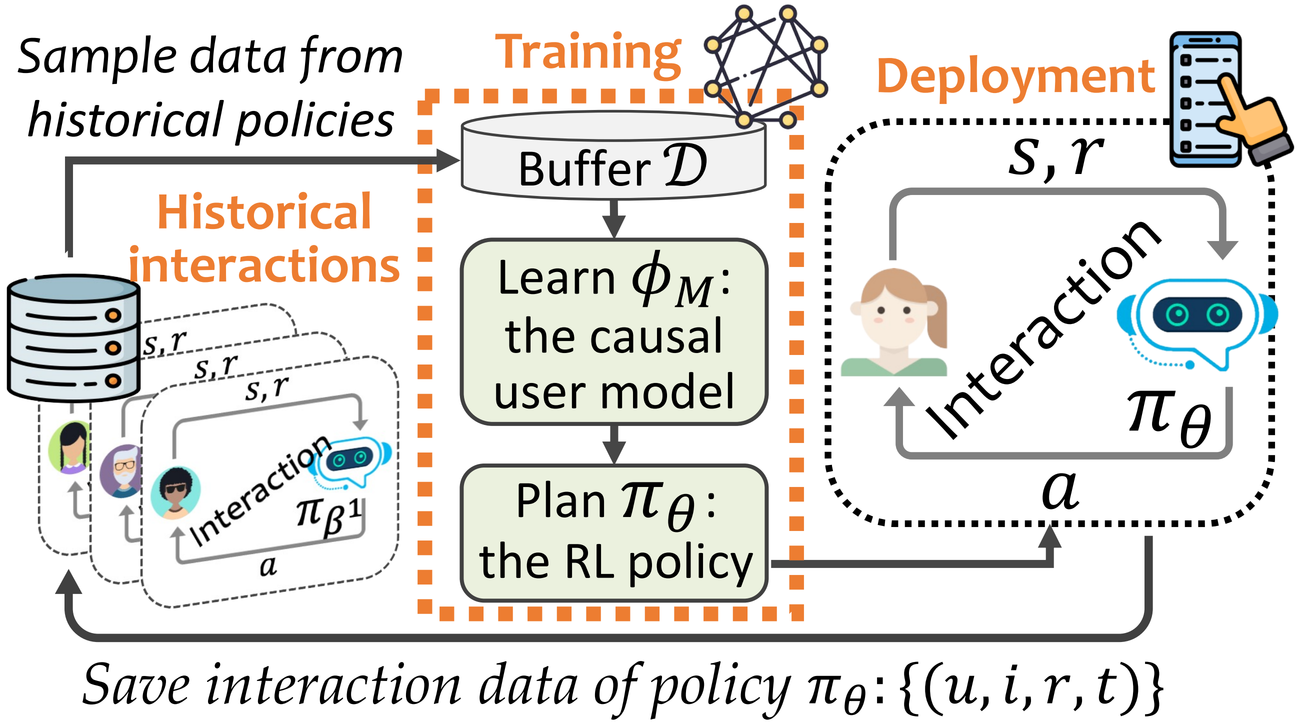
Observation 2: *User satisfaction towards a recommended item drops as the time interval between two similar items reduces.*

1.4 Motivation of the idea

- Propose an unbiased **causal user model** ϕ_M in the model-based **offline reinforcement learning** (RL) framework to **disentangle** the intrinsic user interest from the **overexposure effect** of items.



Traditional online interactive recommender



Counterfactual IRS (CIRS) based on offline RL learning

❖ Outline

1. Background and Motivation.
2. Related Works and Existing Problems
    - Efforts to Mitigate Filter Bubbles
    - Causal Inference-based Recommendation
    - Offline Learning for Online Recommenders
3. Proposed Method: CIRS
4. Experiments

2.1 Efforts to Mitigating Filter Bubbles

Existing Efforts

Help users

- Improve awareness of diverse social opinions *(Gao et al. IUI' 18), (Donkers et al. RecSys' 21)*

Improve the system's

- Diversity *(Aridor et al, RecSys' 20) (Tommasel et al, RecSys' 21)*
- Serendipity *(Xu et al. TKDD' 20)*
- Fairness *(Masrour et al. AAAI' 20)*

Study on

- Whether the failed system can be cured by watching debunking content *(Tomlein et al. RecSys' 21 Best Paper Award)*

However, these efforts mainly focus on the solutions in the static setting, where the effect of filter bubbles is hard to observe and evaluate.

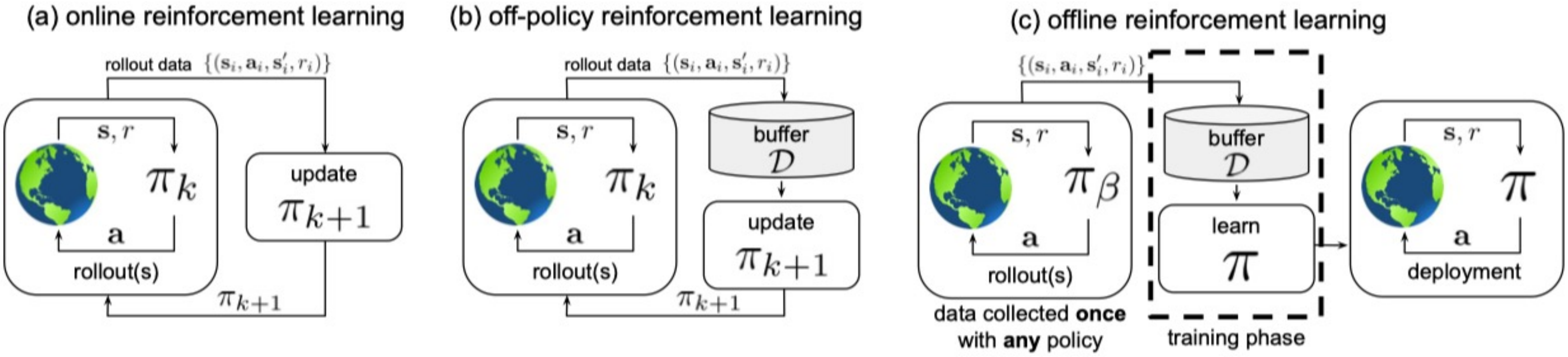
2.2 Causal Inference-based Recommendation

- Causal Inference (CI) has been widely used in NLP, CV, RS
- Instead of exploiting the correlation between input and output, CI explicitly models the causal mechanism among variables.
- General procedures *(Judea Pearl, The Book of Why: The New Science of Cause and Effect)*

1. Construct a structure causal model (SCM) to describe the causal relationship among the related variables.
2. Fit an unbiased model (e.g., implemented as a neural network) based on the SCM on the training data set.
3. In the inference stage, we actively change certain input variables (called intervention) and predict the unbiased result of the target variable.

2.3 Offline Learning for Online Recommenders

- Static model is inflexible. **Reinforcement learning (RL)** introduces a policy that has the ability to adapt to the changing environment. However, it is impractical to train RL online. Because:
  1. for the model, the online interaction with humans is too slow.
  2. for users, interacting with a half-baked system can hurt experiences.
- Solution: **Offline Reinforcement Learning**.



Sergey Levine et al. 2020, Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems

2.3 Offline Learning for Online Recommenders

- Solution: **Offline Reinforcement Learning**.
  - **Off-Policy Evaluation via Importance Sampling:**
    - Main idea: Evaluate the target policy using historical policies.
    - Problem: **High variance**

$$
\begin{aligned}
J(\pi_\theta) &= \mathbb{E}_{\tau\sim\pi_\beta(\tau)}\left[\frac{\pi_\theta(\tau)}{\pi_\beta(\tau)}\sum_{t=0}^{H}\gamma^t r(\mathbf{s},\mathbf{a})\right] \\
&= \mathbb{E}_{\tau\sim\pi_\beta(\tau)}\left[\left(\prod_{t=0}^{H}\frac{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\pi_\beta(\mathbf{a}_t|\mathbf{s}_t)}\right)\sum_{t=0}^{H}\gamma^t r(\mathbf{s},\mathbf{a})\right] \approx \sum_{i=1}^{n} w_H^i \sum_{t=0}^{H}\gamma^t r_t^i,
\end{aligned}
\tag{5}
$$

where $w_t^i = \frac{1}{n}\prod_{t'=0}^{t}\frac{\pi_\theta(\mathbf{a}_{t'}^i|\mathbf{s}_{t'}^i)}{\pi_\beta(\mathbf{a}_{t'}^i|\mathbf{s}_{t'}^i)}$ and $\{(\mathbf{s}_0^i,\mathbf{a}_0^i,r_0^i,\mathbf{s}_1^i,\ldots)\}_{i=1}^n$ are n trajectory samples from $\pi_\beta(\tau)$

  - **Model-based Method:**
    - Main idea: estimate the environment, i.e., transition probability $T(s_{t+1}|s_t,a_t)$
    - Problem: distribution shift, or **biases** in the estimated model.

Sergey Levine et al. 2020, Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems

2.3 Offline Learning for Online Recommenders

- **Summarize RSs according to three dimensions**

  (1) whether the system explicitly builds a ***user model****,*

  (2) whether the system considers ***debiasing****, and*

  (3) whether the system has an ***RL-based policy****.*

Table 1: Six Types of Recommender Systems

	User Model	Debiasing	RL-based	Publications
Static RS	✓			[14, 15, 21]
Unbiased static RS	✓	✓		[23, 24, 39, 41, 52, 61, 64, 66]
Traditional IRS			✓	[25, 26, 28, 55, 60, 65, 67, 69]
Model-based IRS	✓		✓	[4, 8, 53, 62, 63, 68]
OPE-based IRS		✓	✓	[6, 18, 19, 30, 32, 47, 54]
Unbiased model-based IRS	✓	✓	✓	[7, 16] **CIRS (Ours)**

❖ Outline

1. Background and Motivation.
2. Related Works and Existing Problems
3. Proposed Method: CIRS
    - Problem Definition
    - Framework of CIRS
    - Causal Inference-based User Satisfaction Disentanglement
4. Experiments

3.1 Problem Definition

Symbol Definition

- $\mathcal{U}, \mathcal{I}$: the user set and the item set.
- $\mathcal{D}_u = \{\mathcal{S}_u^1, \mathcal{S}_u^2, \cdots, \mathcal{S}_u^{|\mathcal{D}_u|}\}$: The set of all interaction sequence of a user $u \in \mathcal{U}$.
- $\mathcal{S}_u^k = \{(u, i_l, t_l)\}_{1 \le l < |\mathcal{S}_u^k|}$ is the k-th interaction sequence (i.e., trajectory), where user u begins to interact with the system at time t_1 and quits at time $t_{|\mathcal{S}_u^k|}$. $i_l \in \mathcal{I}$ is the recommended item at time t_l.
- $\mathbf{e}_u \in \mathbb{R}^{d_u}, \mathbf{e}_{\boldsymbol{i}} \in \mathbb{R}^{d_i}$: the representation vectors of user u and item i.

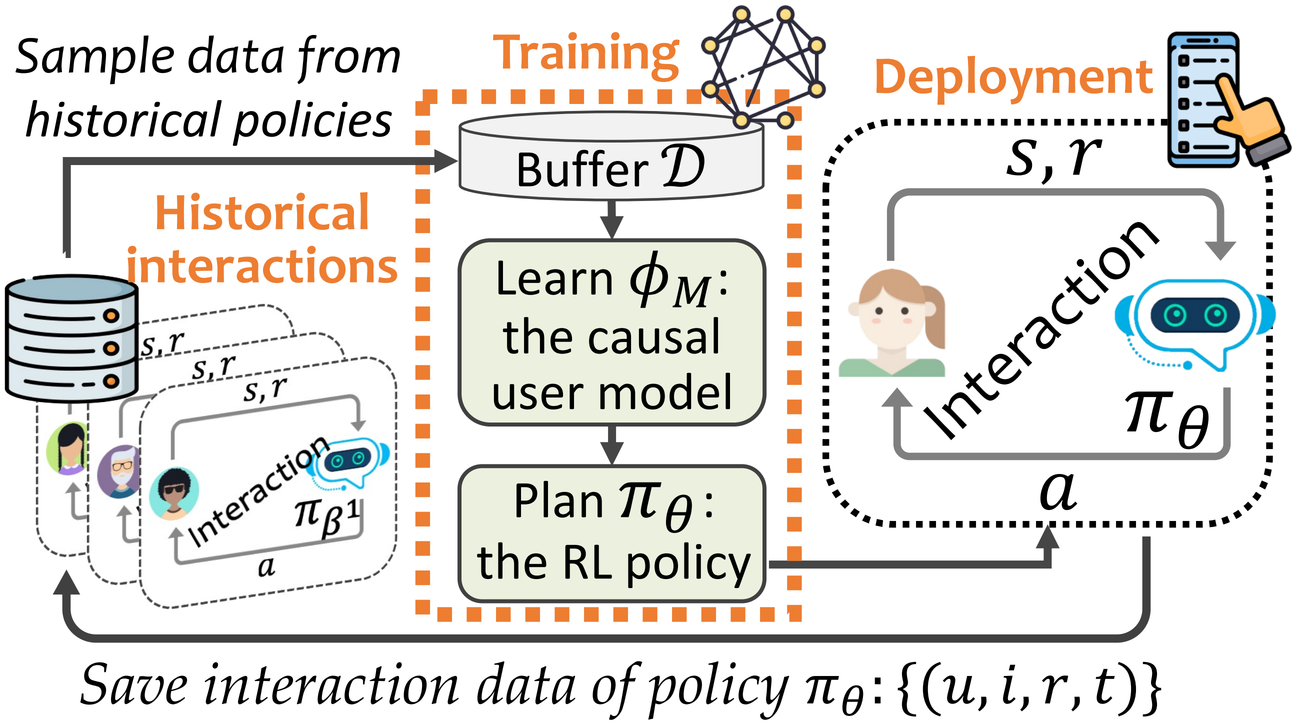
3.1 Problem Definition

Reinforcement learning problem:

- **State:** $\mathbf{s}_t \in \mathbb{R}^{d_s}$ at time t is regarded as a vector representing information of all historical interactions of user u.
- **Action**: The system makes an action a_t at time is to recommend an item to the user. Let $\mathbf{e}_a \in \mathbb{R}^{d_a}$ be the representation vector. In this paper, $\mathbf{e}_a = \mathbf{e}_i$.
- **Reward**: A user u returns feedback as a reward score r reflecting its satisfaction after receiving a recommended item i.
- **Policy network:** $\pi_\theta = \pi_\theta(a_t|s_t)$ is the target policy that decides how to make an action a_t conditioned on the state s_t. In this paper, we use the Proximal Policy Optimization (PPO) algorithm as the policy network.

3.1 Problem Definition

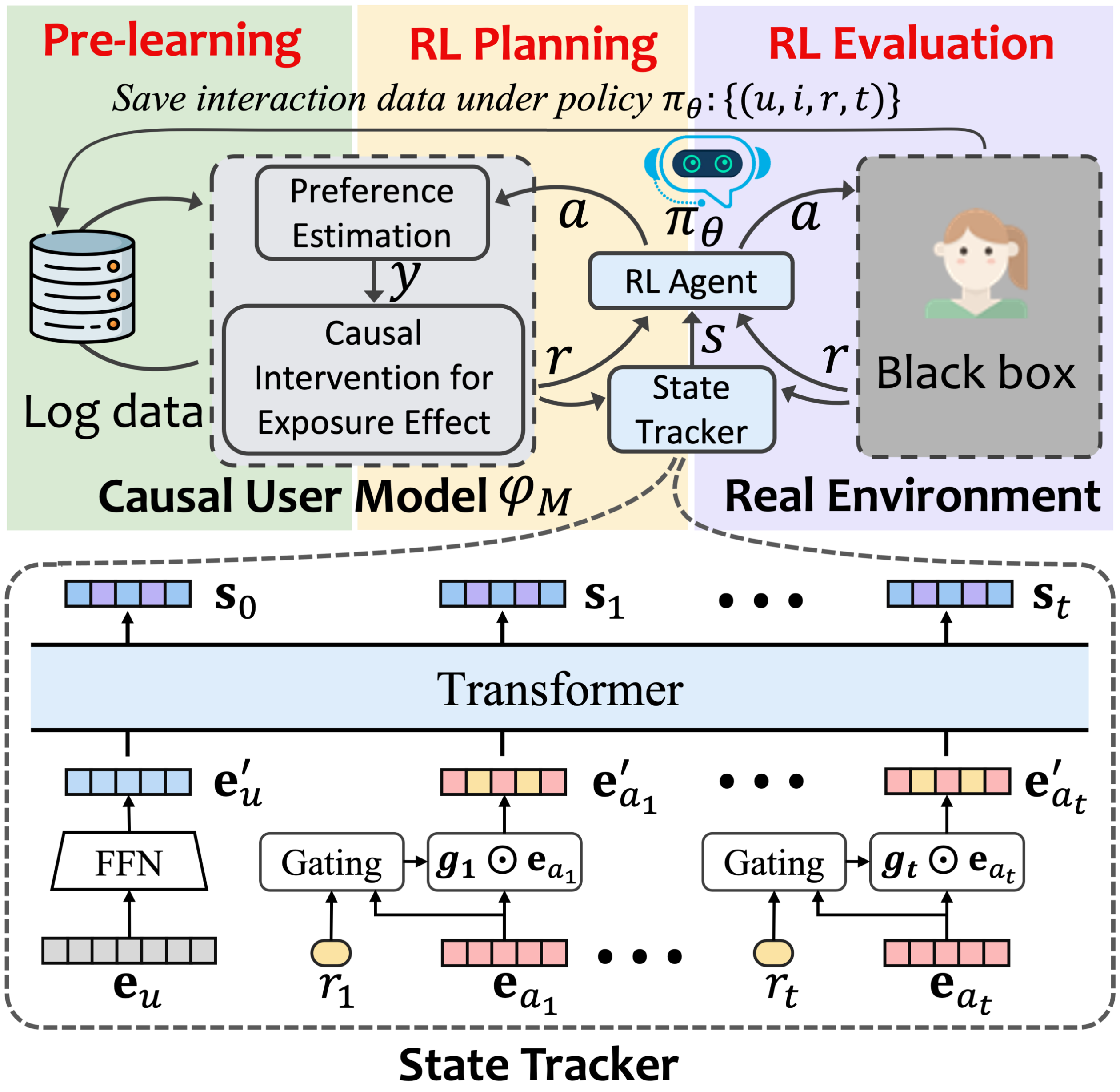
The whole procedure



Counterfactual IRS (CIRS) based on offline RL learning

1. Train a user model ϕ_M via supervised learning on historical data $\{(u, i, r)\}$.
2. Using the learned user model ϕ_M to train policy π_θ. Each time the policy π_θ makes an action (i.e., a recommended item), the causal user model ϕ_M provides a ***counterfactual reward*** r. If π_θ have made similar recommendations before, ϕ_M shrinks the reward r.
3. Serving the learned policy π_θ to users and evaluating the results in the real environment.

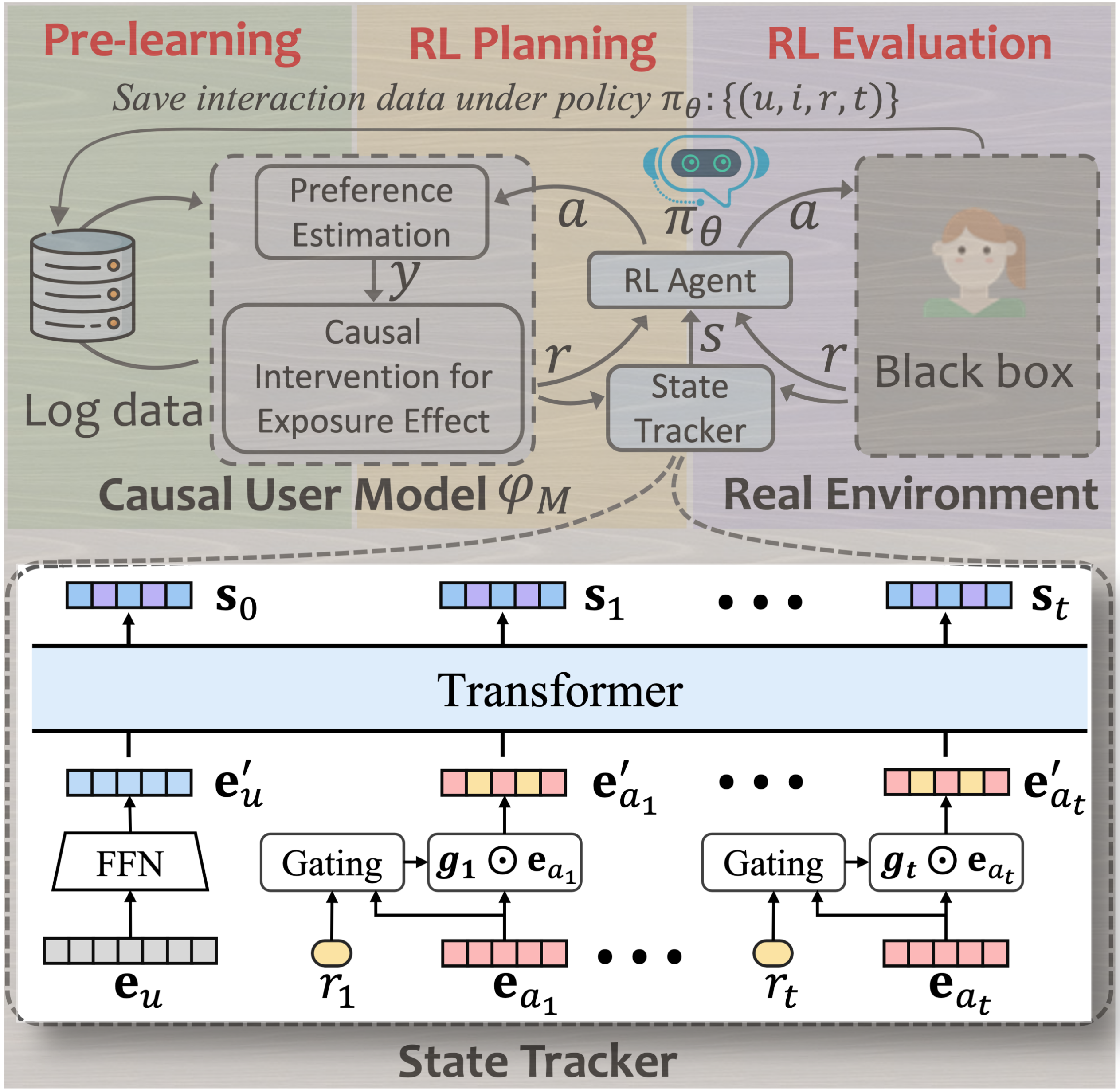
3.2 Framework of CIRS



Three modules in CIRS

- Causal User Model ϕ_M
  - Preference estimation
  - Causal intervention
- RL Policy π_θ
  - Interactive strategy
- State Tracker
  - Recording interaction context

3.2 Framework of CIRS



Transformer-based State Tracker

The states are derive from:

- **User representation**:

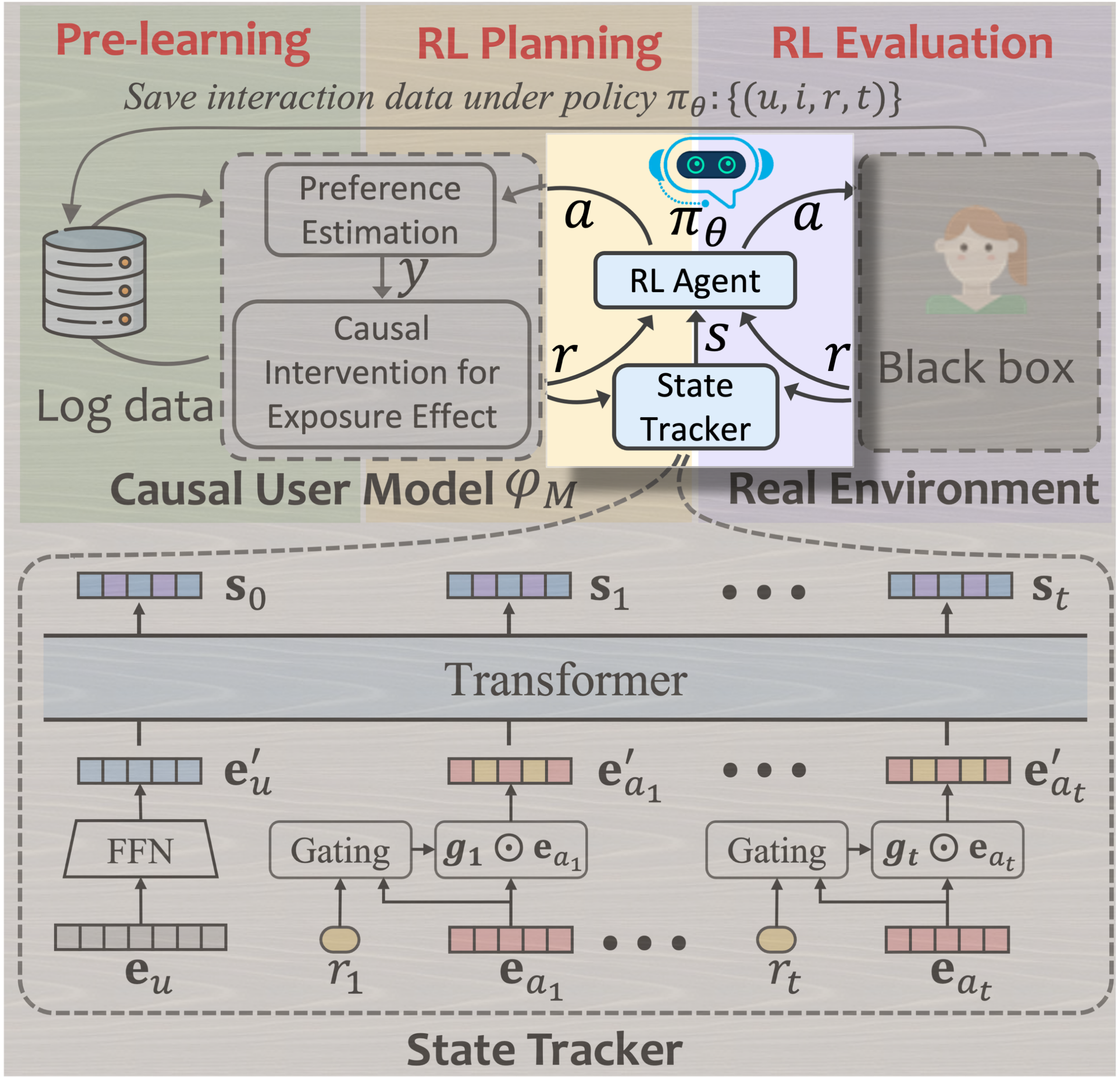
$$\mathbf{e}'_u = \mathrm{FFN}(\mathbf{e}_u)$$

- **Action representation** obtained from a gate mechanism:

$$\mathbf{e}'_{a_t} = \boldsymbol{g}_t \odot \mathbf{e}_{a_t},$$

$$\text{where } \boldsymbol{g}_t = \sigma\left(\boldsymbol{W} \cdot Concat\left(\boldsymbol{r}_t, \boldsymbol{e}_{a_t}\right) + \boldsymbol{b}\right)$$

3.2 Framework of CIRS



RL-based Interactive Recommendation Policy

- π_θ: **PPO algorithm**, an on-policy policy gradient method in RL.

- **Maximize the objective**:

$$\mathbb{E}_t\left[\min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t, clip\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right].$$

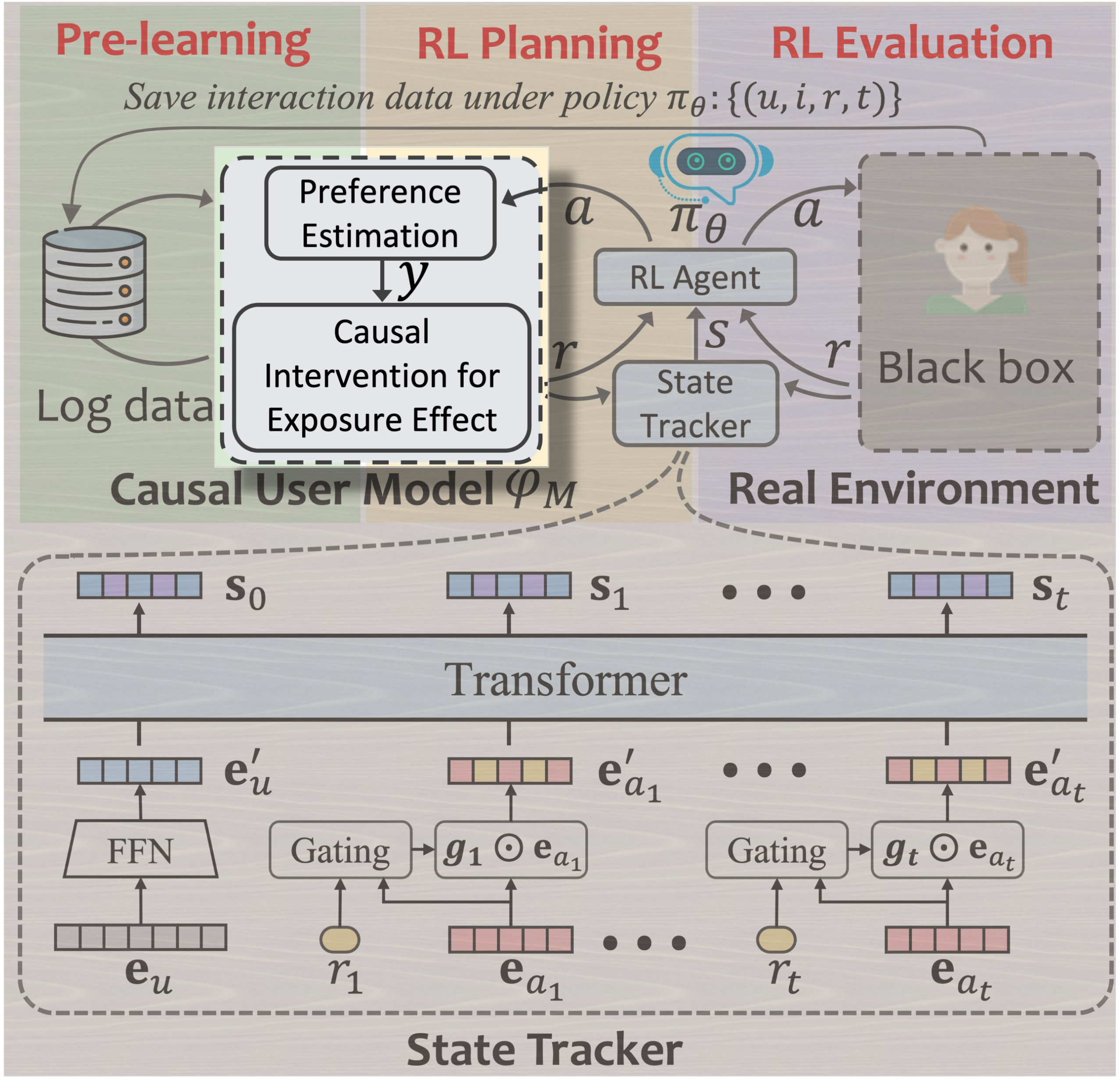
$\pi_{\theta_{old}}$: the policy generating the data

π_θ: the updating policy

$\hat{A}_t$: the advantage function

Schulman et al. Proximal Policy Optimization Algorithms. arXiv:1707.06347 (2017)

3.2 Framework of CIRS

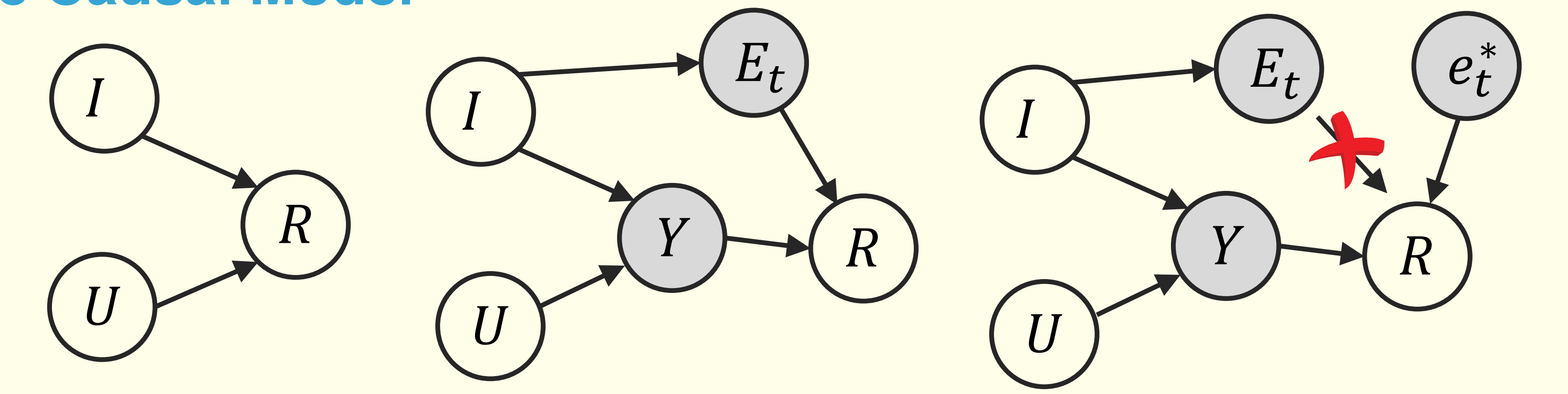


Causal User Model

1. **Estimate user preference** using a naive recommendation model, e.g., *DeepFM*.
2. **Disentangle** the intrinsic user interest from the overexposure effect of items.

3.3 Causal Inference-based User Satisfaction Disentanglement

Structure Causal Model



(a) Causal graph in traditional RSs

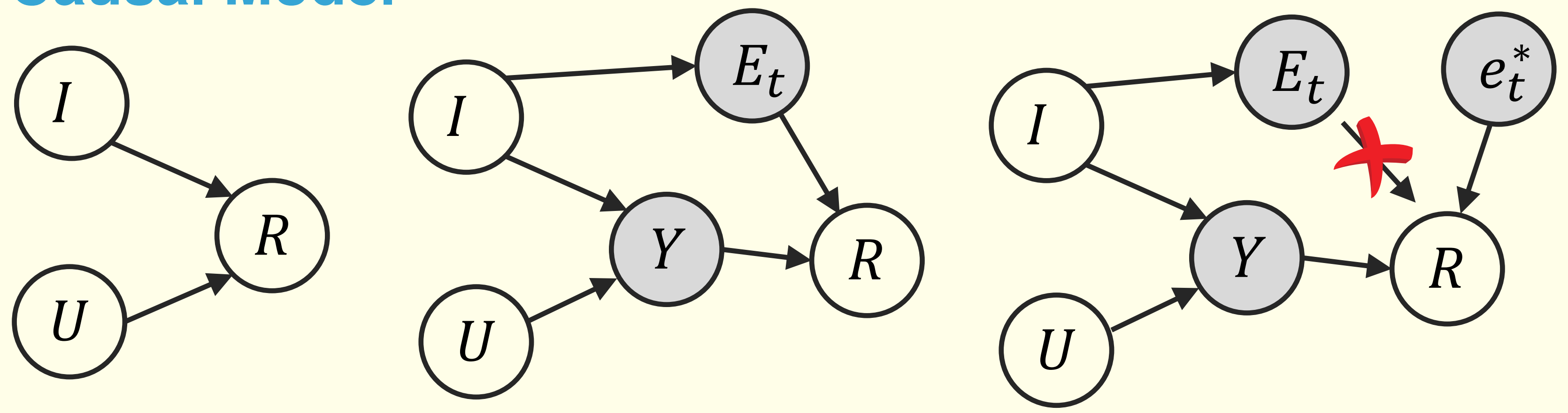
(b) Causal graph in our CIRS

(c) Performing intervention on E_t during inference

- U: a certain user u, e.g., an ID or the profile feature that can represent the user.
- I: an item i that is recommended to user u.
- R: the **user satisfaction**, also used as the ***reward***.
- Y: **intrinsic user interest** (regardless of item exposure)
- E_t: the **overexposure effect** of item i on user u. Where e_t^* is the value of E_t computed in the inference stage (RL planning stage).

3.3 Causal Inference-based User Satisfaction Disentanglement

Structure Causal Model



(a) Causal graph in traditional RSs

(b) Causal graph in our CIRS

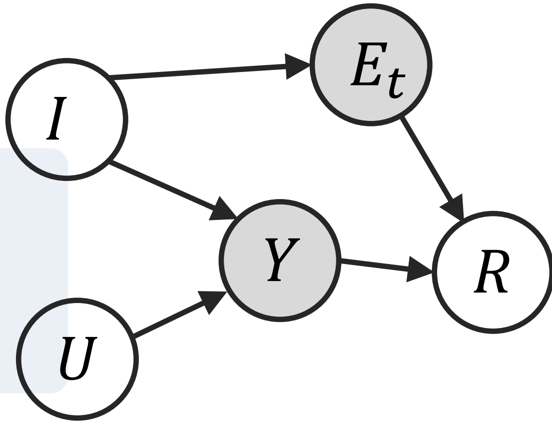
(c) Performing intervention on E_t during inference

Two paths in (b):

$(\boldsymbol{U}, \boldsymbol{I}) \rightarrow \boldsymbol{Y} \rightarrow \boldsymbol{R}$: projects user and item features into the corresponding preference $\hat{y}_{ui} = f_\theta(u, i)$, which can be implemented by various recommendation models (*DeepFM*).

$\boldsymbol{I} \rightarrow \boldsymbol{E_t} \rightarrow \boldsymbol{R}$: represents the real-time overexposure effect e_t^* of an item i on user u that eventually results in the user satisfaction r.

3.3 Causal Inference-based User Satisfaction Disentanglement



Definition of overexposure effect E_t

$$e_t = e_t(u,i) = \alpha_u \beta_i \sum_{\{(u,i_l,t_l)\in S_u^k, t_l<t\}} \exp\left(-\frac{t-t_l}{\tau} \times dist(i,i_l)\right)$$

- $\mathcal{S}_u^k = \{(u, i_l, t_l)\}_{1\le l<|\mathcal{S}_u^k|}$ is the k-th interaction sequence (i.e., trajectory) of user u.
- $dist(i, i_l)$: is distance between two items i and i_l.
- α_u: represents the *sensitivity* of user u to the overexposure effect
- β_i: represents the *unendurableness* of item i.

Definition of user satisfaction $\hat{r}_{ui}^t$

$$\hat{r}_{ui}^t = \frac{\hat{y}_{ui}}{1 + e_t(u,i)}$$

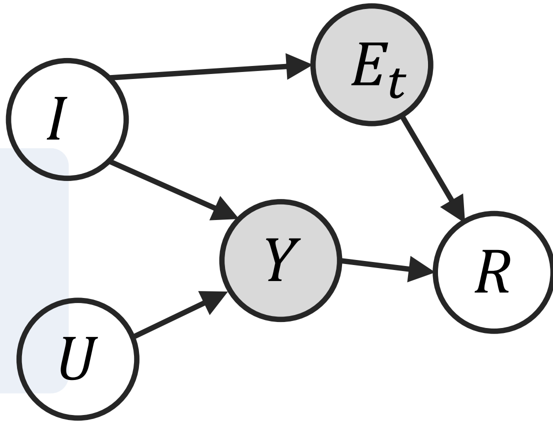
Loss function in training user model:

$$L_{BPR} = -\sum_{\{(u,i,t)\in D, j\sim p_n\}} \log(\sigma(\hat{r}_{ui}^t - \hat{r}_{uj}^t))$$

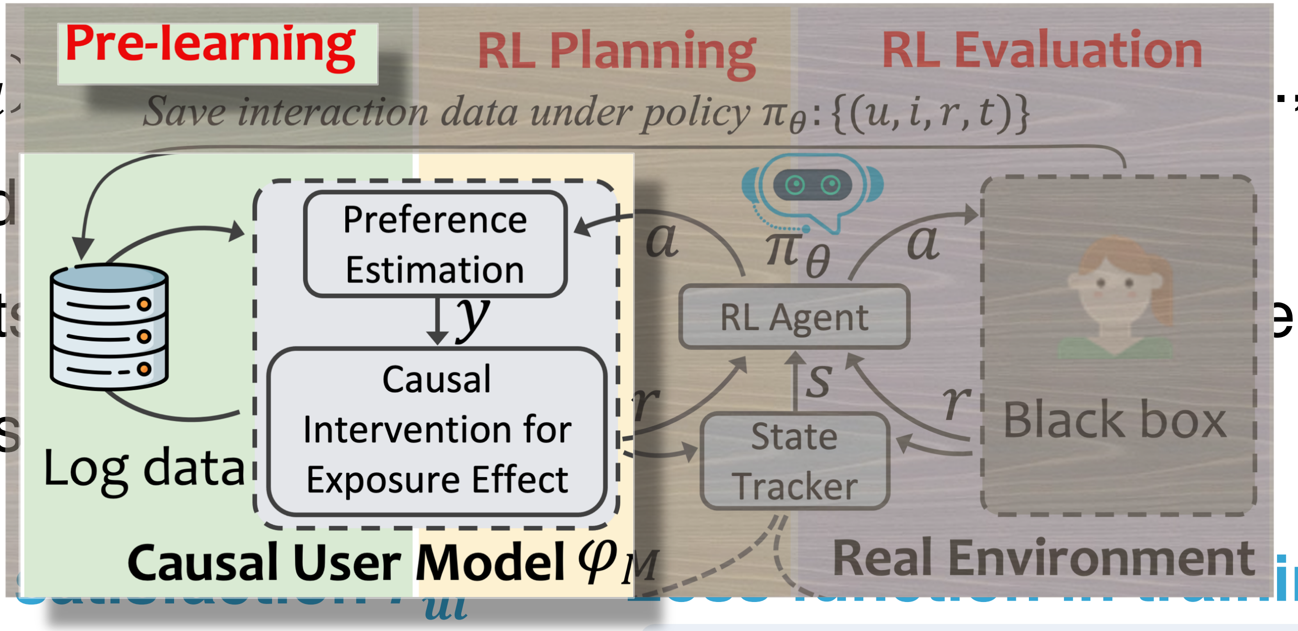
3.3 Causal Inference-based User Satisfaction Disentanglement

Definition of overexposure effect E_t

$$e_t = e_t(u,i) = \alpha_u \beta_i \sum_{\{(u,i_l,t_l)\in S_u^k, t_l<t\}} \exp\left(-\frac{t-t_l}{\tau} \times dist(i,i_l)\right)$$



- $S_u^k = \{(u, i_l, t_l$ ... ., trajectory) of user u.
- $dist(i, i_l)$: is d
- α_u: represent ... e effect
- β_i: represents



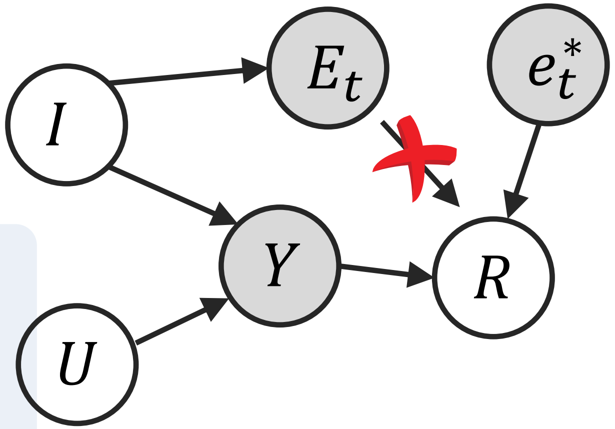
Definition of user ... $_{ui}$ ... ing user model:

$$\hat{r}_{ui}^t = \frac{\hat{y}_{ui}}{1 + e_t(u,i)}$$

$$L_{BPR} = -\sum_{\{(u,i,t)\in D, j\sim p_n\}} \log(\sigma(\hat{r}_{ui}^t - \hat{r}_{uj}^t))$$

3.3 Causal Inference-based User Satisfaction Disentanglement

Causal Intervention on Overexposure Effect

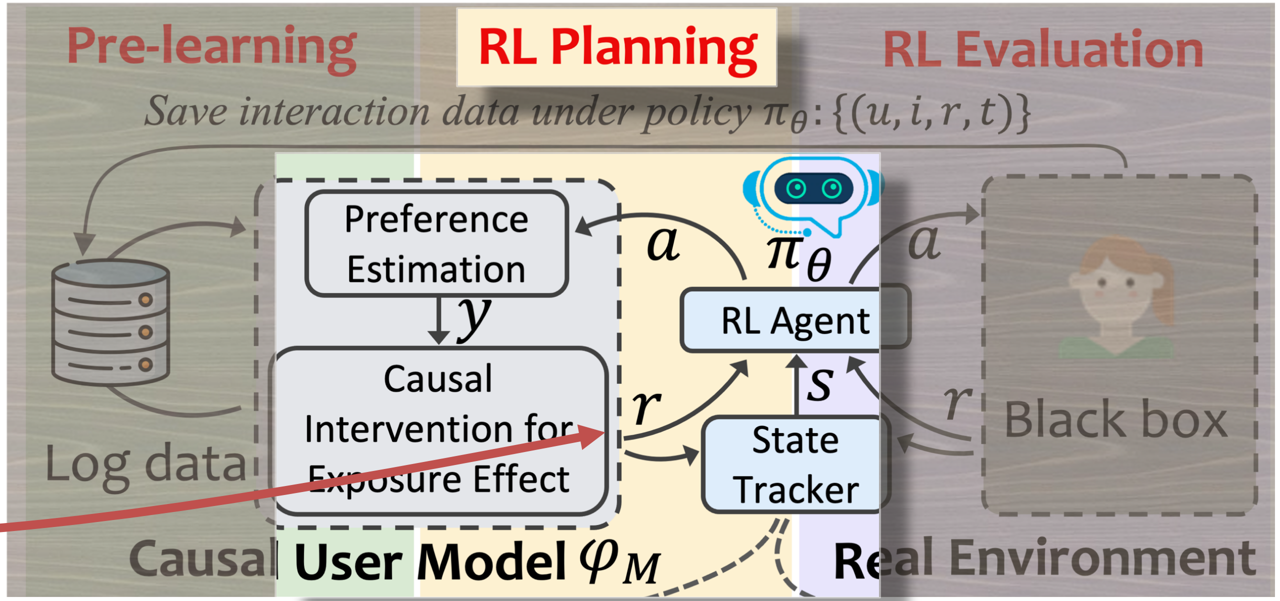


$$e_t^* = \gamma^* \cdot \alpha_u \beta_i \sum_{\{(u,i_l^*,t_l^*) \in S_u^*, t_l^* < t\}} \exp\left(-\frac{t - t_l^*}{\tau^*} \times dist(i, i_l^*)\right)$$

Variables with Asterisk " $*$ " are these in the inference stage (i.e., RL planning stage)

Adjusted user satisfaction $\hat{r}_{ui}^{t*}$

$$\hat{r}_{ui}^{t*} = \frac{\hat{y}_{ui}}{1 + e_t^*(u, i)}$$



❖ Outline

1. Background and Motivation.
2. Related Works and Existing Problems
3. Proposed Method: CIRS
4. Experiments
   - Experimental Setup
   - Performance Comparison
   - More Analysis

4.1 Experimental Setup

Recommendation Environments: *1. VirtualTaobao*

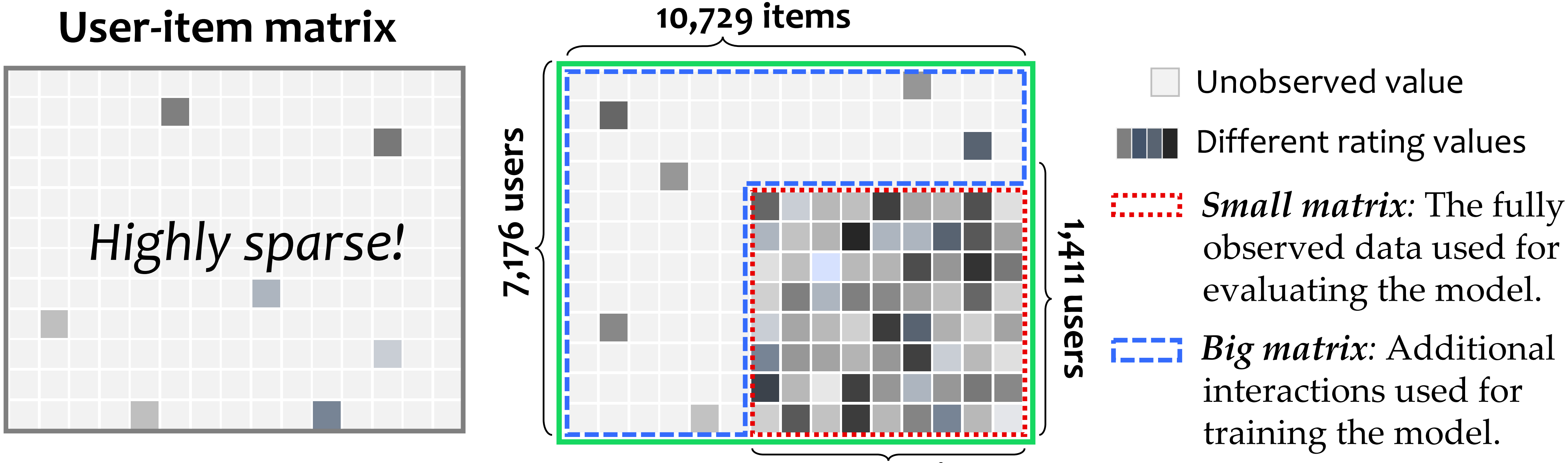
- A benchmark RL environment for recommendation.
- Created by simulating the behaviors of real users on Taobao.
- A user is represented as an 88-dimensional vector $\mathbf{e}_u \in \{0,1\}^{88}$
- An item is represented as a 27-dimensional vector $\mathbf{e}_{\boldsymbol{i}} \in \mathbb{R}^{27}, 0 \leq \mathbf{e}_{\boldsymbol{i}} \leq 1$.
- When a recommender makes an action, the environment will immediately return a *reward* representing the number of clicks, $r \in \{0,1,\cdots,10\}$.

4.1 Experimental Setup

Recommendation Environments: 2. KuaiEnv

	#users	#Items	#Interactions	Density
Small matrix	1,411	3,327	4,676,570	99.6%
Big matrix	7,176	10,729	12,530,806	13.4%

Item feature:	Each video has at least 1 and at most 4 tags out of the totally 31 tags, e.g., {Sports}.
Social network:	*Small matrix*: 146 users have friends. *Big matrix*: 472 users have friends.



(a) Traditional recommendation datasets

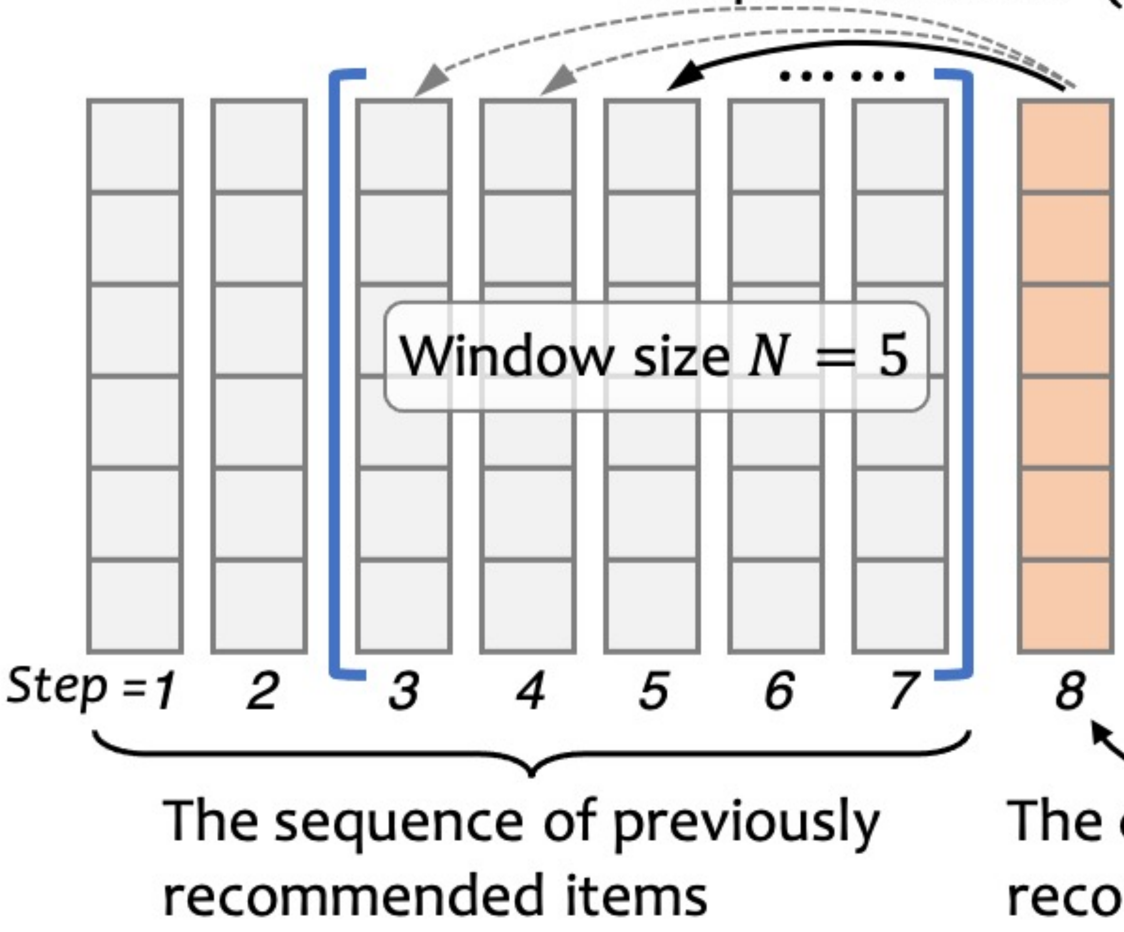
(b) The User-item matrices in the proposed *KuaiRec*

4.1 Experimental Setup

Exit Mechanism:

Compute distance (*VirtualTaobao*: Euclidean distance,

KuaishouEnv: Check if there is a overlapped attribute)



Feature vectors of items

- *VirtualTaobao*: 27-dim continuous vectors
- *KuaishouEnv*: 31-dim multi-hot vectors

Exit mechanism:

- *VirtualTaobao*: Quit if any distance lower than d_Q.
- *KuaishouEnv*: Quit if more than n_Q items have overlapped attributes.

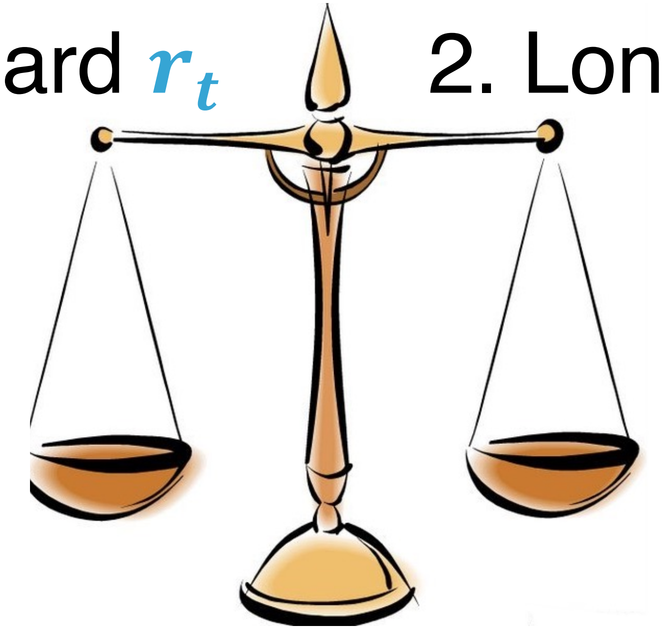
4.1 Experimental Setup

Evaluation metric:

Accumulated reward = $\sum_{t=1}^{T} r_t$, *which requires:*

1. High single-round reward r_t

2. Long trajectory length T



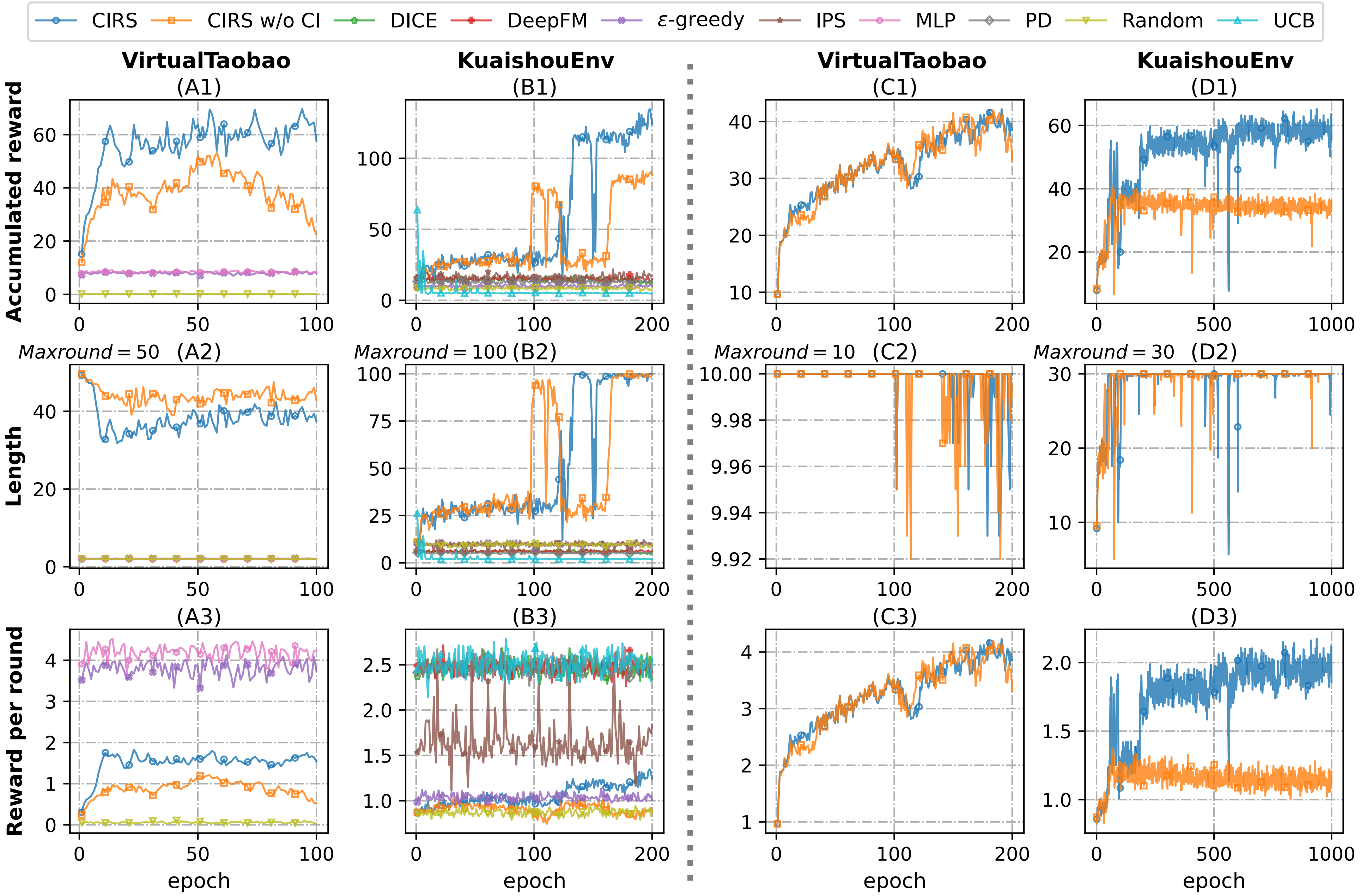
Find a balance

4.1 Experimental Setup

Baselines *(Recommendation Model + Policy)*

- **DeepFM** *(DeepFM + Softmax Sampling)*
- **IPS** *(IPS + Softmax Sampling)*
- **PD** *(PD + Softmax Sampling)*
- **DICE** *(DICE + Softmax Sampling)*
- **MLP** *(MLP + SoftMax Sampling)*
- **Random**
- **ϵ-greedy** *(DeepFM + ϵ-greedy)*
- **UCB** *(DeepFM + UCB)*
- **CIRS** *(User Model + PPO)*
- **CIRS w/o CI** *（CIRS without causal inference module）*

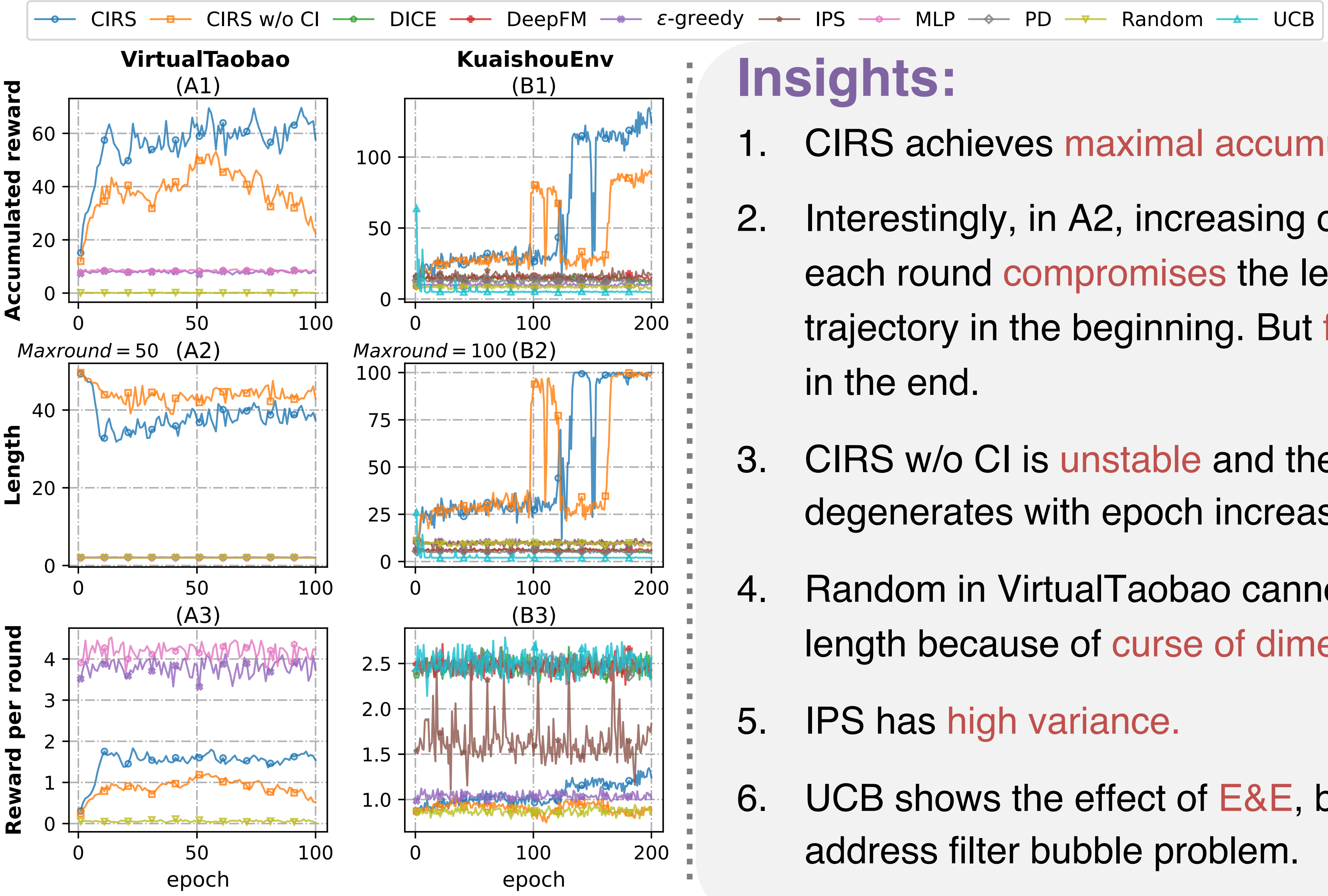
4.2 Results



(A-B) Results with large interaction rounds

(C-D) Results with limited interaction rounds

4.2 Results



(A-B) Results with large interaction rounds

Insights:

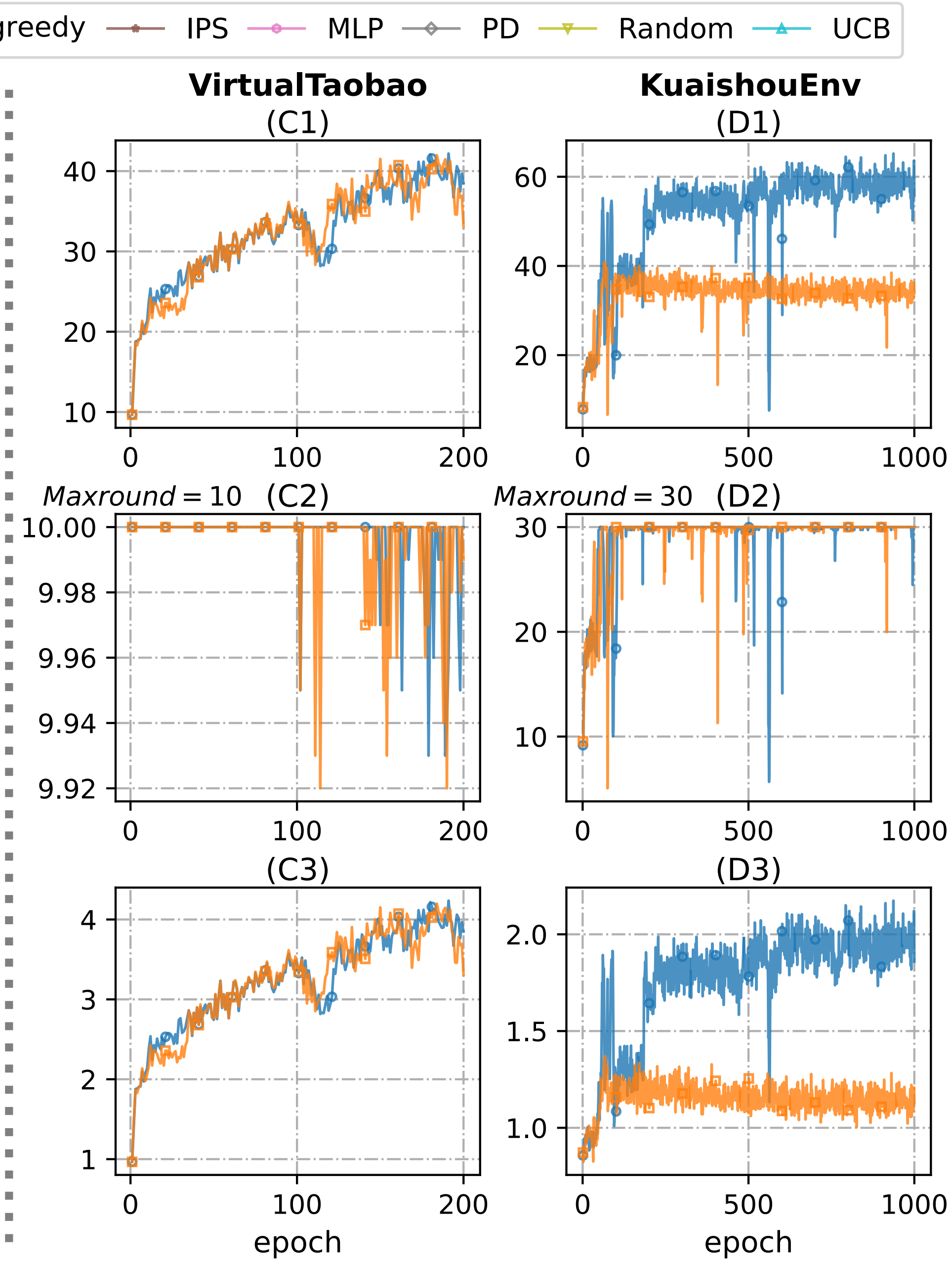
1. CIRS achieves maximal accumulated reward.
2. Interestingly, in A2, increasing of the reward in each round compromises the length of trajectory in the beginning. But finds a balance in the end.
3. CIRS w/o CI is unstable and the performance degenerates with epoch increasing.
4. Random in VirtualTaobao cannot bring longer length because of curse of dimensionality.
5. IPS has high variance.
6. UCB shows the effect of E&E, but it cannot address filter bubble problem.

4.2 Results

Insights:

In VirtualTaobao, both policies achieve the same level of single-round performance as the static methods.

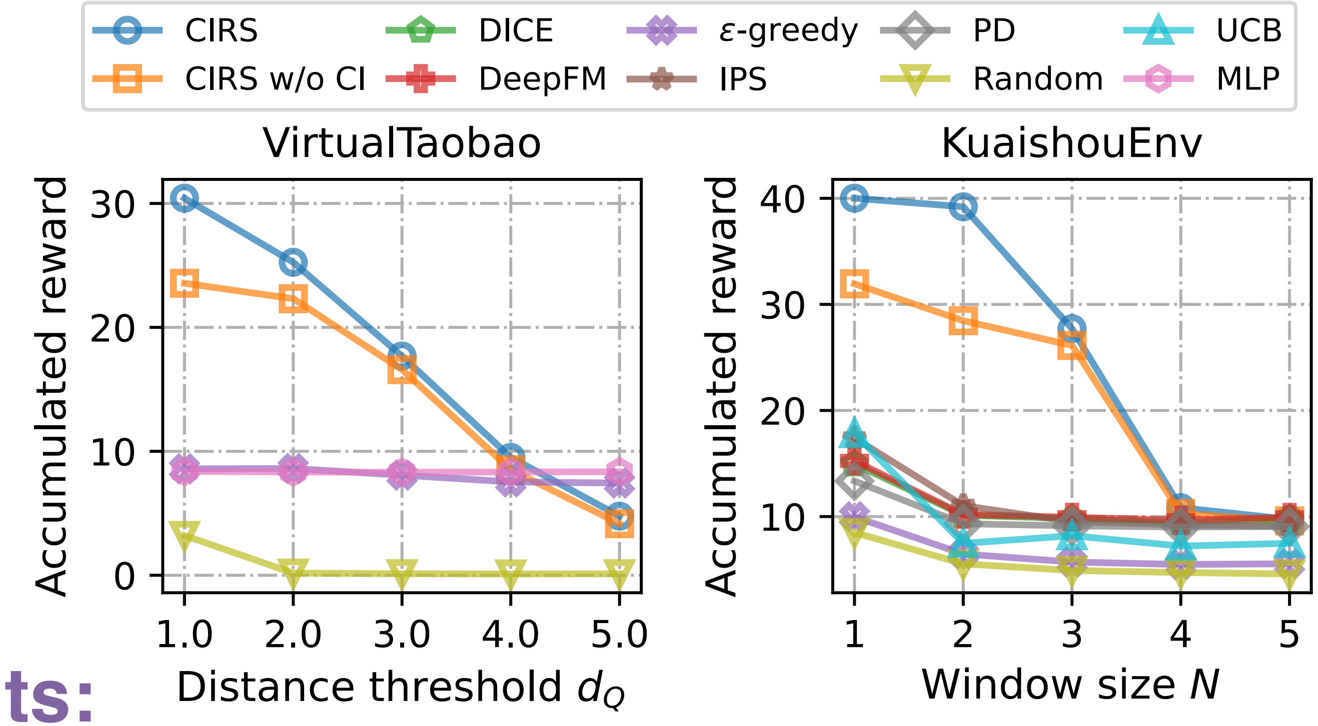
In KuaishouEnv, Armed with causal inference, CIRS beats its counterpart greatly.



(C-D) Results with limited interaction rounds

4.2 Results

Results under different user sensitivity

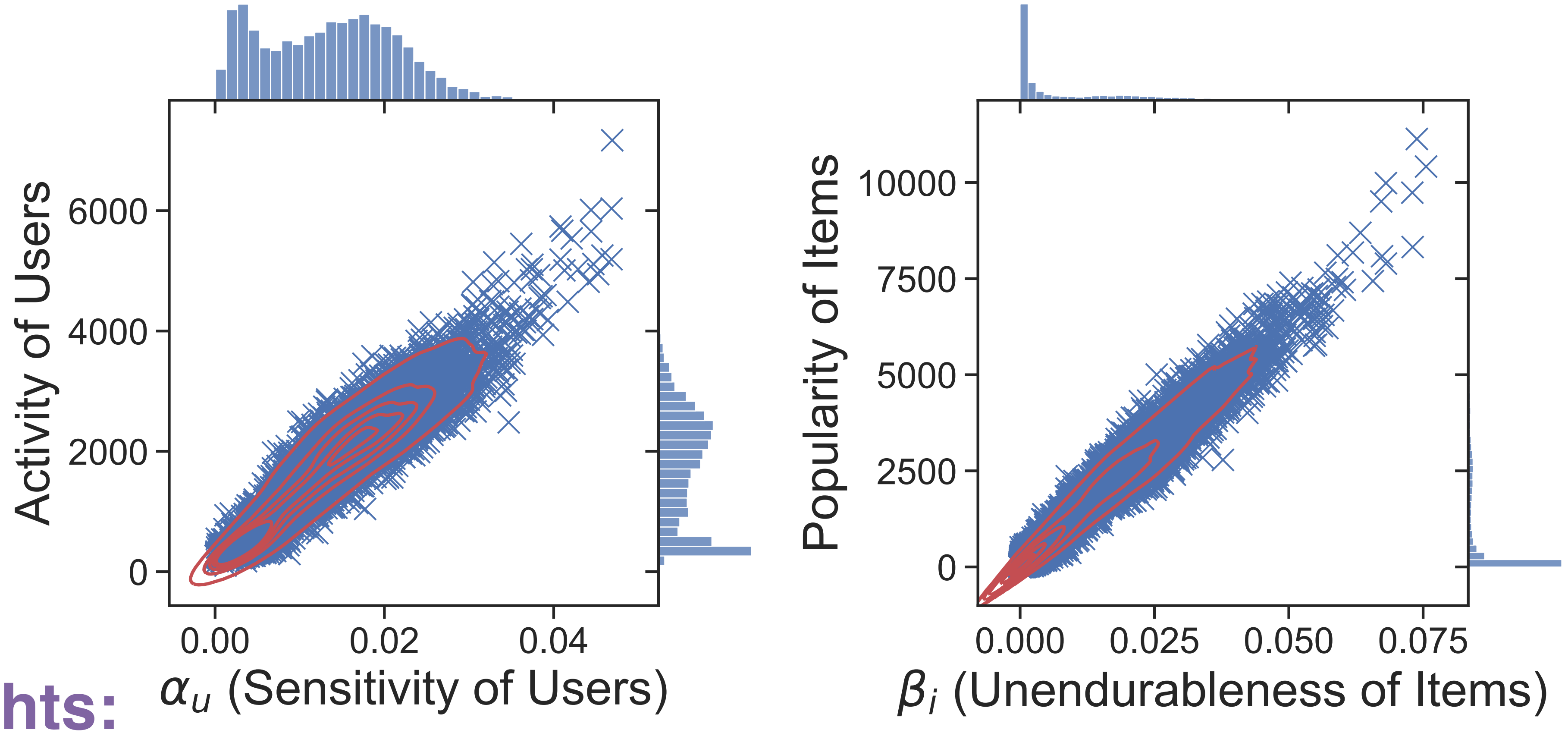


Insights:

- When users become more sensitive, the performance of CIRS and CIRS w/o CI drop.
- Other baselines are not suitable in addressing filter bubbles.

4.3 Analysis Effect of Key Parameters

$$e_t = e_t(u,i) = \alpha_u \beta_i \sum_{\{(u,i_l,t_l)\in S_u^k, t_l<t\}} \exp\left(-\frac{t-t_l}{\tau} \times dist(i,i_l)\right)$$

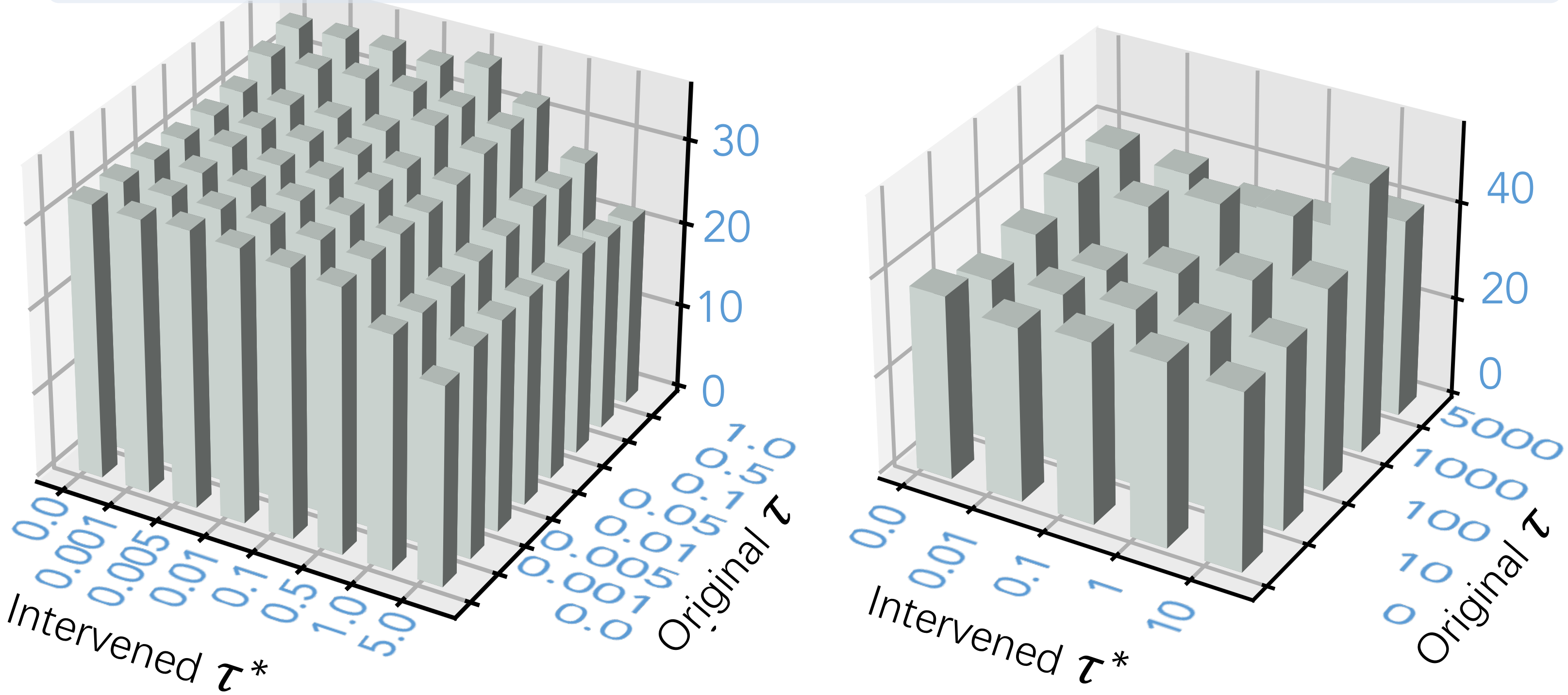


Insights:

- An active user is easier to get bored when viewing overexposed videos.
- Popular videos are less endurable when they are overexposed.

4.3 Analysis **Effect of Key Parameters**

$$e_t^* = \gamma^* \cdot \alpha_u \beta_i \sum_{\{(u, i_l^*, t_l^*) \in S_u^*, t_l^* < t\}} \exp\left(- \frac{t - t_l^*}{\tau^*} \times dist(i, i_l^*) \right)$$



Insights:

- Suitable (τ, τ^*) pair indeed improve the performance.
- The orders of magnitude of τ and τ^* are different because the unit of time is different, i.e., second(s) vs. step(s).

❖ Conclusion

1. The first work for learning to **burst filter bubbles** in interactive recommendation, where filter bubbles can be observed and evaluated.
2. Proposed the **CIRS** based on offline reinforcement learning. We are the first to utilize the **causal inference in interactive recommendation**.
3. Collected a **fully filled dataset** (density: 100\%) from Kuaishou to create an interactive recommendation environment.
4. The **experiments** show that our proposed model can burst filter bubbles and gain the maximal accumulative rewards.

中国科学技术大学
University of Science and Technology of China

快手
KUAISHOU TECHNOLOGY

Thanks

Chongming Gao | 高崇铭
chongming.gao@gmail.com