Reinforcement Learning for Short Video Recommender Systems

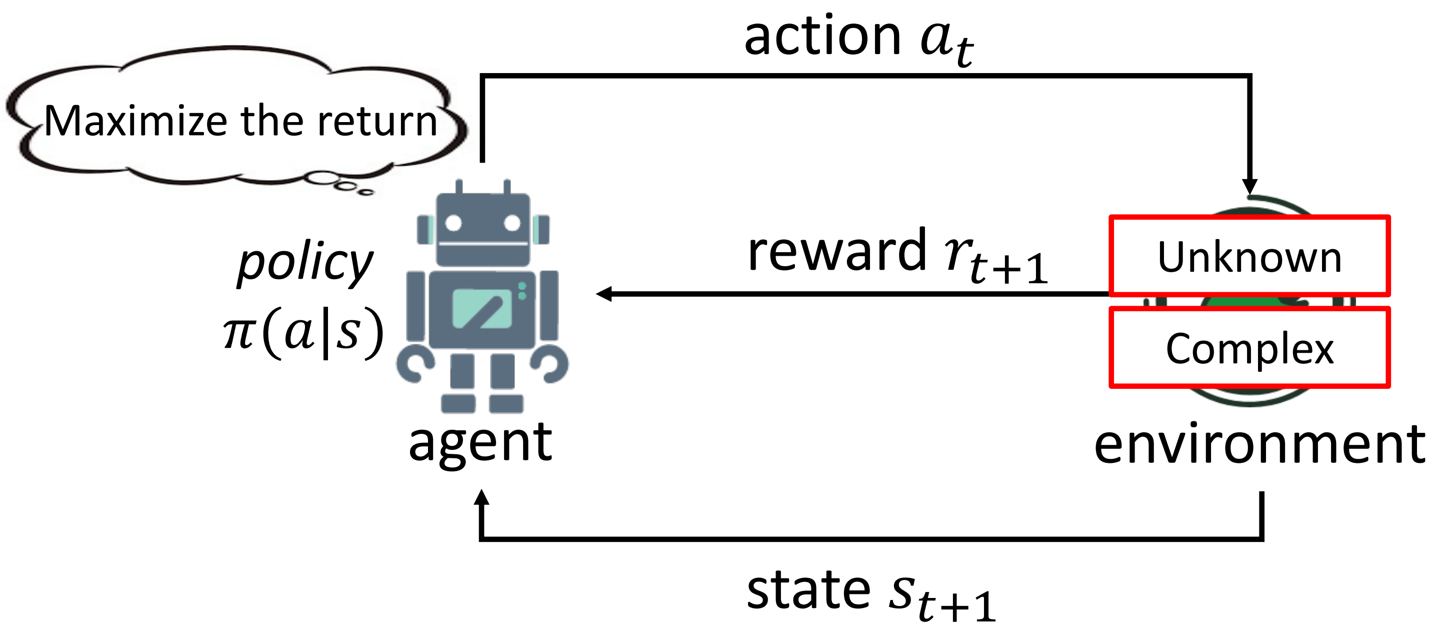
Qingpeng Cai

Outline

1. Reinforcement Learning
2. Reinforcement Learning for Short Video Recommender Systems
    - Reinforcement Learning for short video RS
    - Advanced: Research works about RL-based short video RS
        - Multi-objectives (WWW 2023)
        - Delayed feedback: retention (WWW 2023)
3. Future Research Directions

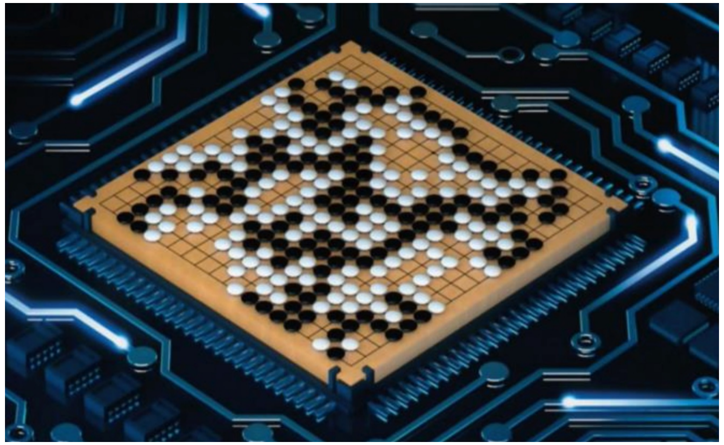
Reinforcement Learning

Reinforcement Learning



Deep Reinforcement Learning

Atari



Go

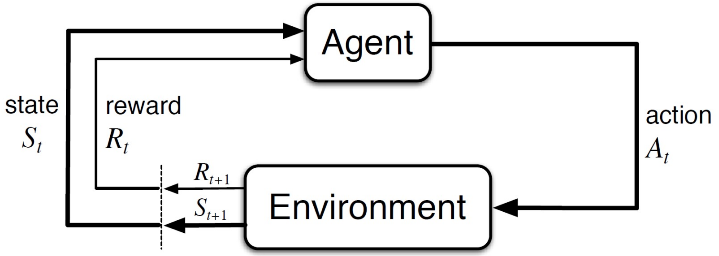
StarCraft II

Robotics

RLHF with PPO

Introduction of Reinforcement Learning

- Agent maximizes rewards by interaction with environments



- Markov Decision Process (MDP)：
- Markov Property： $P(s\_(t+1) \mid s\_t, \ldots, s\_1, a_t) = P(s\_(t+1) \mid s\_t, a_t)$
- Tuple:(S, A, P, R, γ)
- Objective：Find the policy that maximizes the discounted sum of rewards

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Bellman Equation
  - Value function

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \right)$$

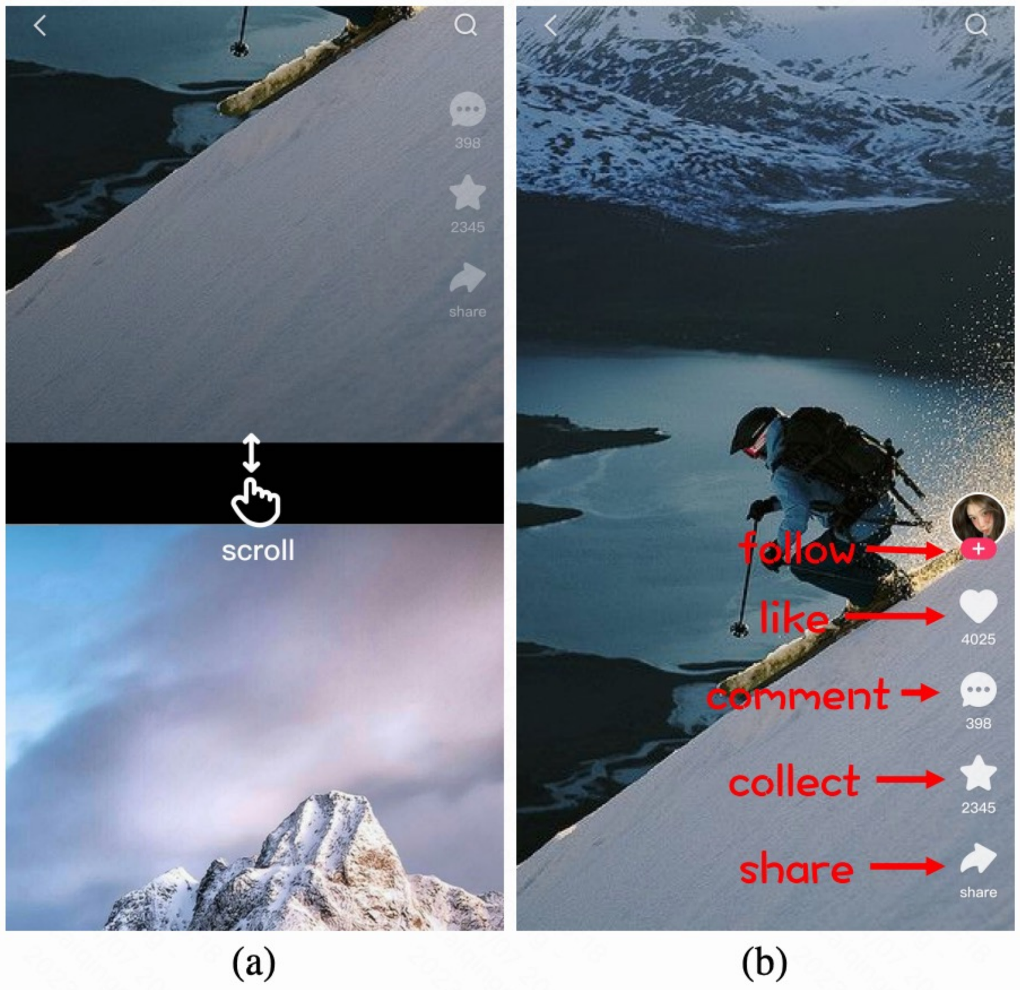
  - Q function

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s', a')$$

Reinforcement Learning for Short Video RS

Difference between Short Video RS and Other RS

- Users interact with short video RS
  - Scroll up and down
  - Watch multiple videos
- Multi-objectives
  - Watch time of multiple videos
    - Main objective, Dense responses
  - Share, Download, Comment
    - Sparse responses, constraints
- Delayed feedback
  - Session depth
  - User Retention



Motivation of RL in Short Video RS

- Problems of supervised learning methods
  - predict the value of an item or a list of items
  - lack of exploration and can not optimize the long-term value

- Hyper-parameter tuning in Kuaishou RS
  - Many hyper-parameters Exist
    - $w_1 * x_1 + w_2 * x_2 + \cdots + w_n * x_n$
  - How to learn optimal parameters w to maximize different objectives?
    - Objectives: watch time, interactions, session depth
  - Non-gradient methods CEM/Bayes are used in Kuaishou
    - Unable to optimize long-term metric
    - Lack of personalization
- RL
  - Exploration
  - Aim to maximize the long-term performance

RL for Hyper-parameter Tuning: MDP

- MDP
  - State:(user information, user history)
    - User information:
    - User history: states, actions, and rewards of previous steps
  - Action
    - Parameters of several ranking functions
    - A continuous vector
  - Reward
    - $r_t = watch\ time + like\ count\ *\ w_{like} + follow\ count\ *\ w_{follow} + forward\ count\ *\ w_{forward}$
  - Episode
    - Requests from opening the app to leaving the app

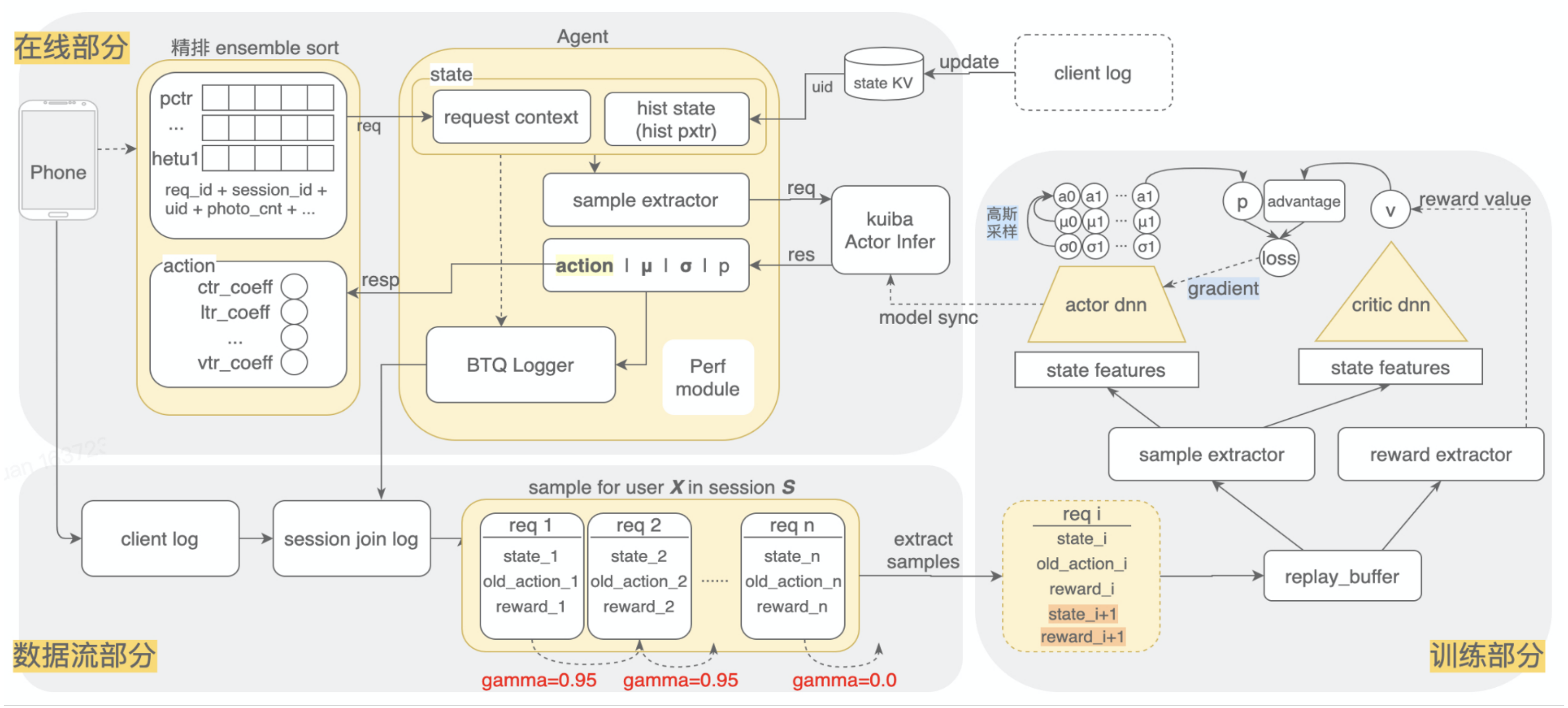
RL for Hyper-parameter Tuning: Algorithms

- Objective

$$\max \sum_{t=0}^{T} \gamma^t (time_t + w1 * like_t + w2 * follow_t + w3 * forward_t + w4 * comment_t + w5 * 0.1)$$

- Policy
  - DNN
  - Input state, output mu and sigma
  - Sample action from Gaussian distribution
- Algorithm Selection
  - Reinforce
    - Slow convergence, only works for single objective
  - PPO
    - On-policy, does not work for off-policy setting of KS
  - A3C
    - Faster convergence，sensitive to different reward coefficient

RL for Hyper-parameter Tuning: Training and Inference



RL for Hyper-parameter Tuning: Live Results

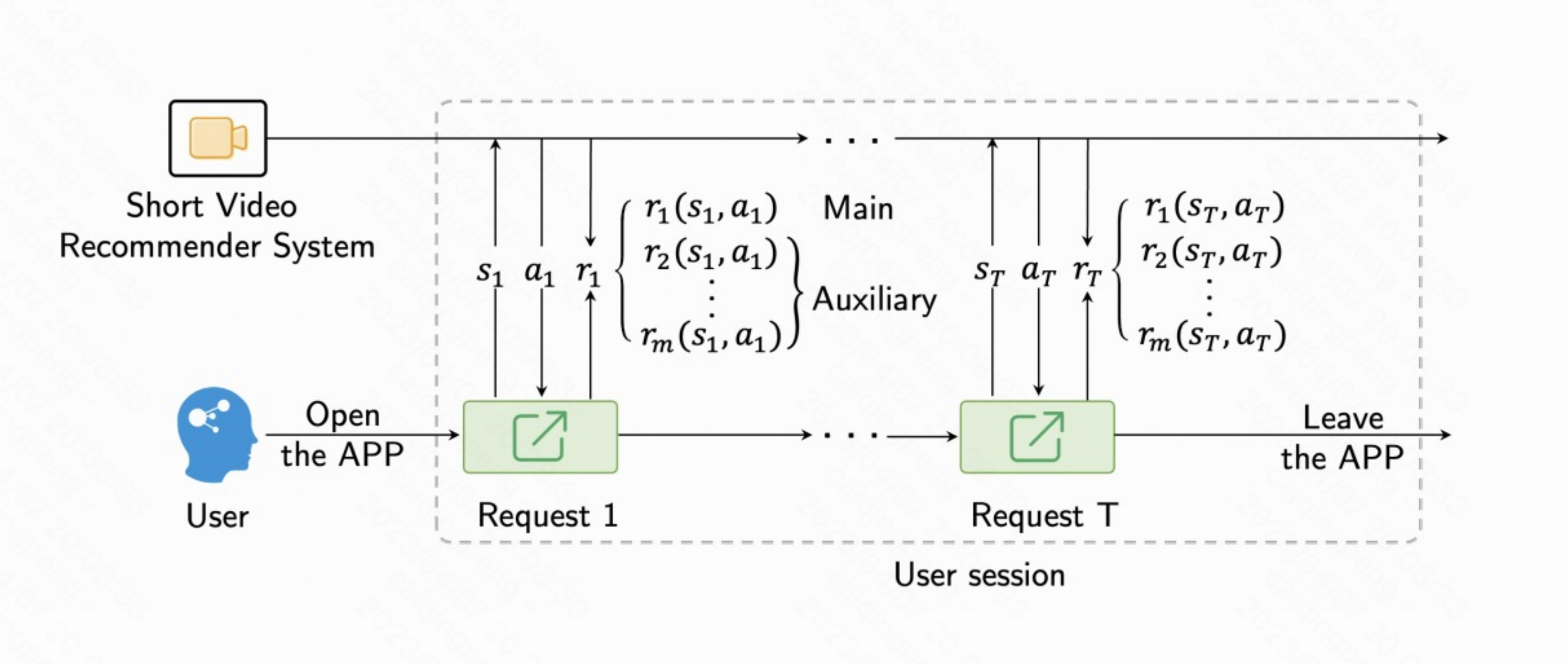
- Loss functions
  - Actor loss $-log\pi(a|s)(r + \gamma * V(s') - V(s))$
  - Critic loss $(r + \gamma * V(s') - V(s))^2$
- Live Experiments
  - Baseline: CEM
  - Avg app time +0.15% Watch time +0.33%
  - Fully launched
- Comparison with Contextual Bandits
  - Gamma=0: contextual bandits
  - Gamma=0.95 compares with gamma=0
    - App time +0.089%, VV +0.37%
    - RL performs better than Bandits!

Challenges of RL for Short Video RS

- Unstable Environment
  - Each user is a environment, rather than fixed game
  - System fluctuates between days and hours
- Multi-objectives
  - Different reward signals in short-videos: dwell time, like, follow, forward, comment, visiting depth
- Safe and efficient exploration
  - Random exploration hurts user experience
- Delayed feedback and credit assignment
  - The long-term engagement signal is delayed and noisy
  - It is hard to allocate credits to immediate actions

RL for Ranking(Multi-objectives, WWW 2023)

Constrained Markov Decision Process (CMDP)



- Env: user
- RS: agent
- Step: each request
- Action: a video
- Immediate Rewards: Watch time and interactions

- The optimization program

$$\max_{\pi} \quad U_1(\pi)$$
$$\text{s.t.} \quad U_i(\pi) \geq C_i, \quad i = 2, \ldots, m,$$

Challenges

- A direct method is learn a policy to optimize its Lagrangian

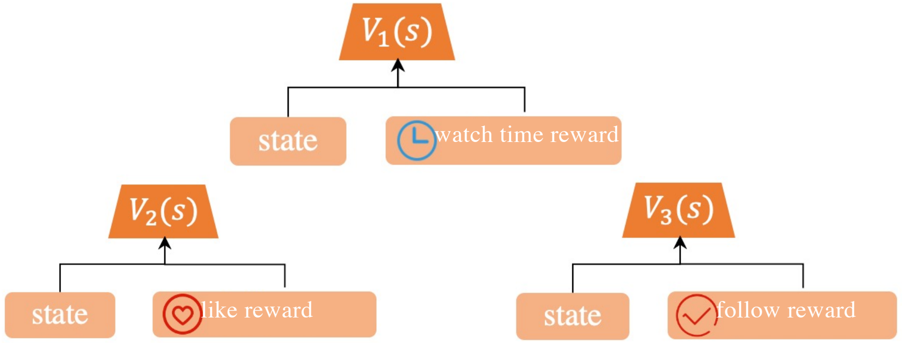
$$\mathcal{L}(\pi, \lambda) = U_1(\pi) + \sum_{i=2}^{m} \lambda_i (U_i(\pi) - C_i), \quad \text{where} \quad \lambda_i \geq 0.$$

- Problem:
  - The estimation is not accurate for sparse signals
    - The dense signal, such as watch time dominates the estimation

  - It is hard to maximize the Lagrangian
    - larger search space due to multiple constraints
    - time costly

Multi-Critic Policy Estimation

- Each critic estimated the value of one objective



- Compare Joint and Separate learning
  - Joint Learning: V_0 learns watch time+interaction
  - Separate Learning: V_1 learns watch time, V_2 learns interaction
  - Use MAE error to estimate two learning method
  - Separate learning outperforms joint learning
    - by 0.191% and 0.143% in terms of both watch time and interaction

Two-Stage Constrained Actor-Critic

- Stage One
  - For each auxiliary response, learn a policy to optimize its cumulative reward

$$\phi_i^{(k+1)} \leftarrow \arg\min_{\phi} \mathbb{E}_{\pi_{\theta_i^{(k)}}} \left[\left(r_i(s,a) + \gamma_i V_{\phi_i^{(k)}}(s') - V_{\phi}(s) \right)^2 \right].$$

We update the actor to maximize the advantage:

$$\theta_i^{(k+1)} \leftarrow \arg\max_{\theta} \mathbb{E}_{\pi_{\theta_i^{(k)}}} \left[A_i^{(k)} \log\left(\pi_{\theta}(a|s) \right) \right]$$

$$\text{where} \quad A_i^{(k)} = r_i(s,a) + \gamma_i V_{\phi_i^{(k)}}(s') - V_{\phi_i^{(k)}}(s).$$

Two-Stage Constrained Actor-Critic

- Stage Two
  - For the main response, learn a policy to optimize its cumulative reward
  - Softly regularize the policy to be close to other auxiliary policies

$$
\begin{aligned}
\max_{\pi} \quad & E_{\pi}[A_1^{(k)}] \\
\text{s.t.} \quad & D_{KL}(\pi || \pi_{\theta_i}) \leq \epsilon_i, \quad i = 2, \ldots, m, \\
\text{where} \quad & A_1^{(k)} = r_1(s, a) + \gamma_1 V_{\phi_1^{(k)}}(s') - V_{\phi_1^{(k)}}(s).
\end{aligned}
$$

Two-Stage Constrained Actor-Critic

- Stage Two
  - For the main response, learn a policy to optimize its cumulative reward
  - Softly regularize the policy to be close to other auxiliary policies

THEOREM 1. *The Lagrangian of Eq. (5) has the closed form solution*

$$\pi^*(a|s) \propto \prod_{i=2}^{m} \left(\pi_{\theta_i}(a|s)\right)^{\frac{\lambda_i}{\sum_{j=2}^{m}\lambda_j}} \exp\left(\frac{A_1^{(k)}}{\sum_{j=2}^{m}\lambda_j}\right), \tag{6}$$

where λ_i with $i = 2, \ldots, m$ are Lagrangian multipliers.

Two-Stage Constrained Actor-Critic

- Stage Two
  - For the main response, learn a policy to optimize its cumulative reward
  - Softly regularize the policy to be close to other auxiliary policies

Given data collected by $\pi_{\theta_1^{(k)}}$, we learn the policy π_{θ_1} by minimizing its KL divergence from the optimal policy π^*:

$$\theta_1^{(k+1)} \leftarrow \arg\min_{\theta} E_{\pi_{\theta_1^{(k)}}}[D_{KL}(\pi^*(a|s)||\pi_\theta(a|s))]$$

$$= \arg\max_{\theta} E_{\pi_{\theta_1^{(k)}}}\left[\frac{\prod_{i=2}^{m}\left(\pi_{\theta_i}(a|s)\right)^{\frac{\lambda_i}{\sum_{j=2}^{m}\lambda_j}}}{\pi_{\theta_1^{(k)}}(a|s)}\exp\left(\frac{A_1^{(k)}}{\sum_{j=2}^{m}\lambda_j}\right)\log\pi_\theta(a|s)\right]. \quad (7)$$

Smaller λ, weaker constraint

Same λ for all objectives

Offline Experiments

Table 2: Performance of different algorithms on KuaiRand.

Algorithm	Click↑	Like↑(e-2)	Comment↑(e-3)	Hate↓(e-4)	WatchTime↑
BC	0.5338	1.231	3.225	2.304	12.85
Wide&Deep	0.5544	1.244	3.344	2.011	12.84
	3.86%	1.07%	3.69%	−12.7%	−0.08%
DeepFM	0.5549*	1.388*	3.310	2.112	12.92
	3.95%*	12.76%*	2.64%	−8.31%	0.53%
RCPO	0.5510	1.386	3.628*	2.951	13.07*
	3.23%	12.57%	12.5%*	28.1%	1.70%*
RCPO-Multi-Critic	0.5519	1.367	3.413	2.108	13.00
	3.41%	11.04%	5.83%	−8.49%	1.14%
Pareto	0.5438	1.171	3.393	0.9915*	11.90
	1.87%	−4.85%	5.22%	−56.96%*	−7.4%
TSCAC	**0.5570**	**1.462**	**3.728**	1.870	**13.14**
	4.35%	**18.80%**	**15.6%**	−18.83%	**2.23%**

The number in the bracket stands for the unit of this column; The number in the first row of each algorithm is the NCIS score.
The percentage in the second row means the performance gap between the algorithm and the BC algorithm.
The numbers with $*$ denote the best performance among all baseline methods in each response dimension.
The last row is marked by bold font when TSCAC achieves the best performance at each response dimension.

Live Experiments

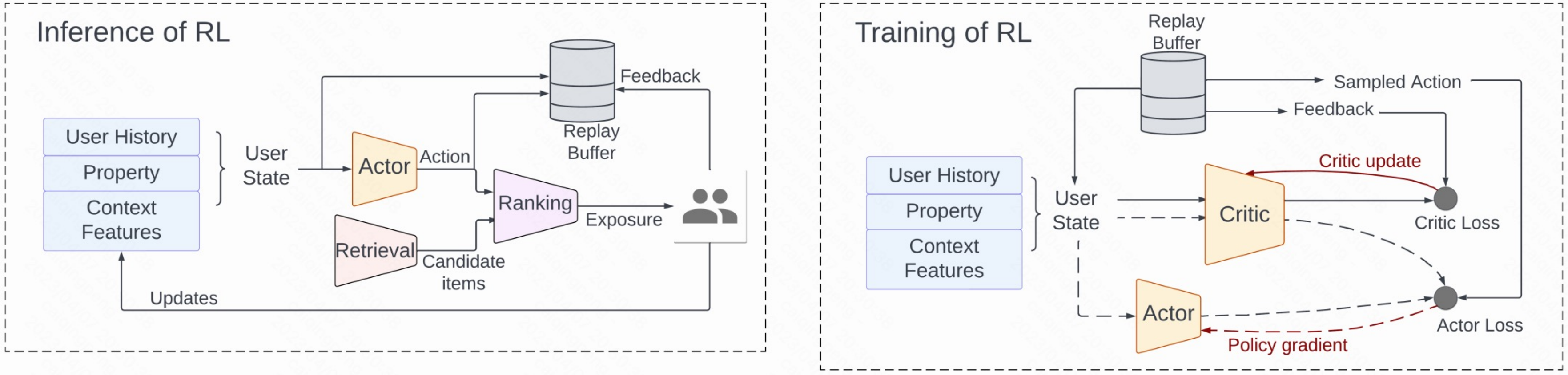


Figure 4: The workflow of RL in production system.

Table 3: Performance comparison of different algorithms with the LTR baseline in live experiments.

Algorithm	WatchTime	Share	Download	Comment
RCPO	+0.309%	−0.707%	0.153%	−1.313%
Interaction-AC	+0.117%	+5.008%	+1.952%	−0.101%
TSCAC	+0.379%	+3.376%	+1.733%	−0.619%

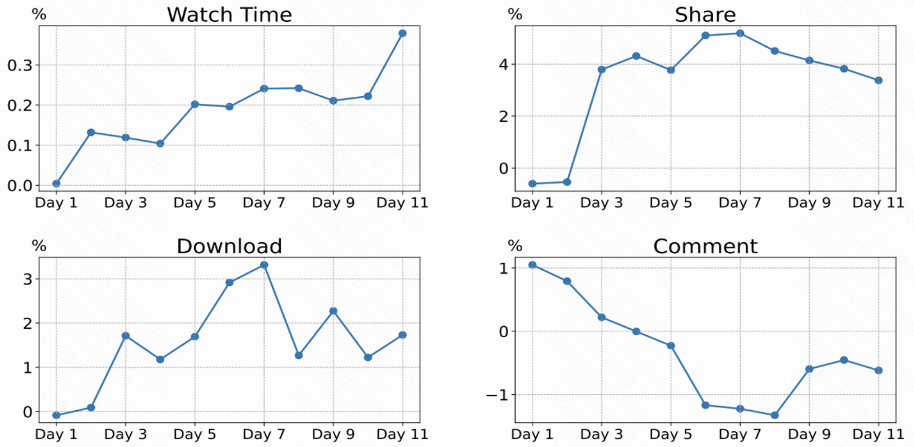
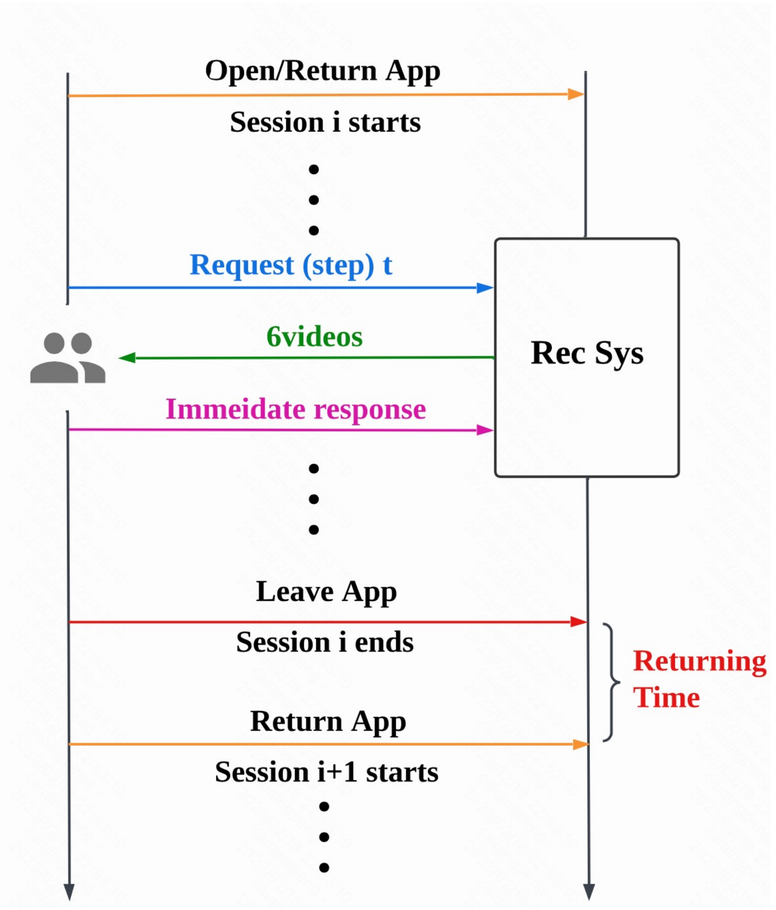


Figure 5: Online performance gap of TSCAC over the LTR baseline of each day.

RL for Hyparameter Tuning(Delayed feedback, WWW 2023)

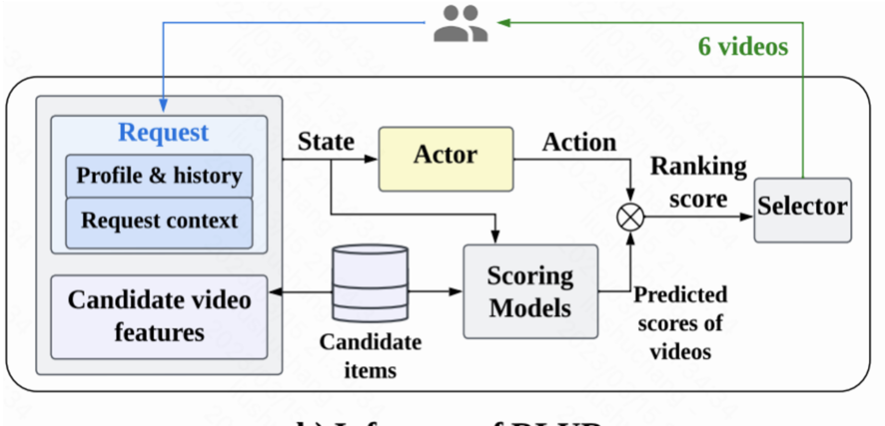
User Retention in Short-video Recommendation

- User Retention
  - Directly affects DAU
  - long-term feedback after multiple requests
    - Hard to decompose, similar to Go
    - Point-wise and list-wise methods can not optimize
- Solution: RL optimizes user retention directly
  - Minimize the cumulative sum of returning time
    - Equal to improving user visits
  - One of the first works to directly optimize user retention
    - Previous works focus on cumulative immediate feedback

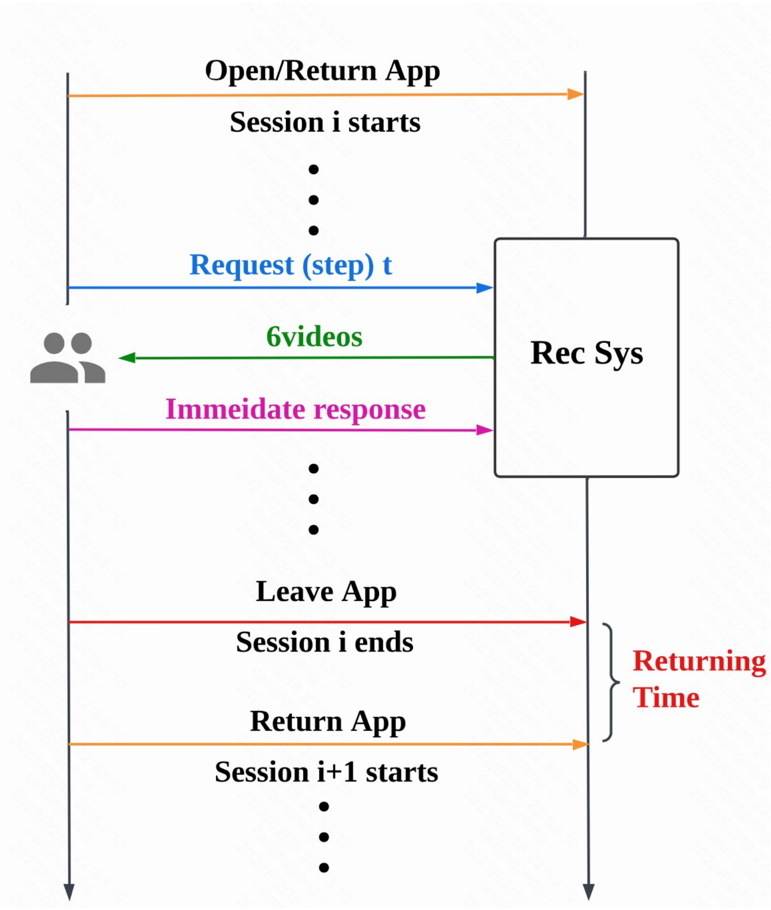


Infinite Horizon Request-based Markov Decision Process

- State
  - User profile, user history, candidate video features
- Action: a vector to ensemble ranking functions



b) Inference of RLUR

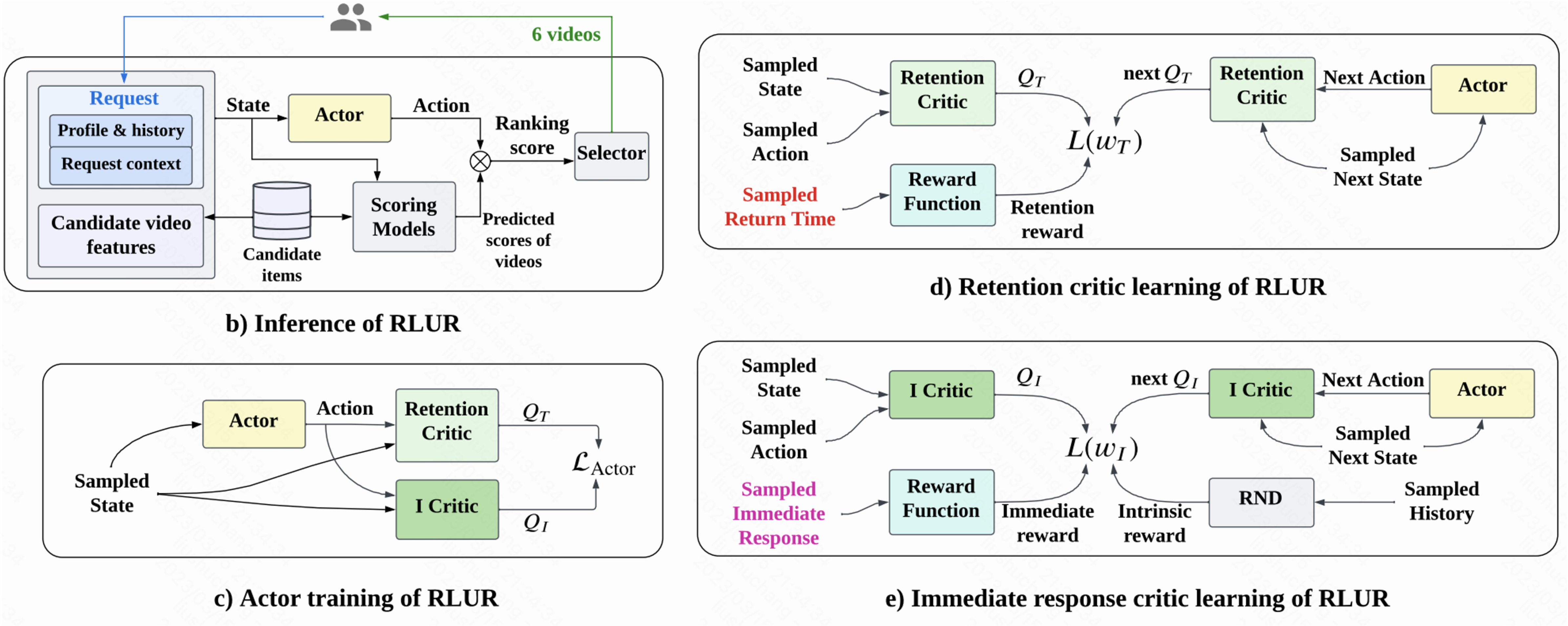


- Immediate Rewards
  - The sum of watch time and interactions, $I(s_{it}, a_{it})$
- Returning time
  - Time gap between the last step of session s_i and the first step of session s_{i+1}
- Objective: minimize $\sum_{i=1}^{\infty} \gamma^{i-1} T(s_i)$

Challenges of Retention

- Uncertainty
  - Retention is not fully decided by the recommendation
  - Affected by social events
- Bias
  - Biased with time and user activity
  - High active users have higher retention and more samples
- Long delay time
  - Retention reward returns in hours to days
  - Cause the instability of online RL

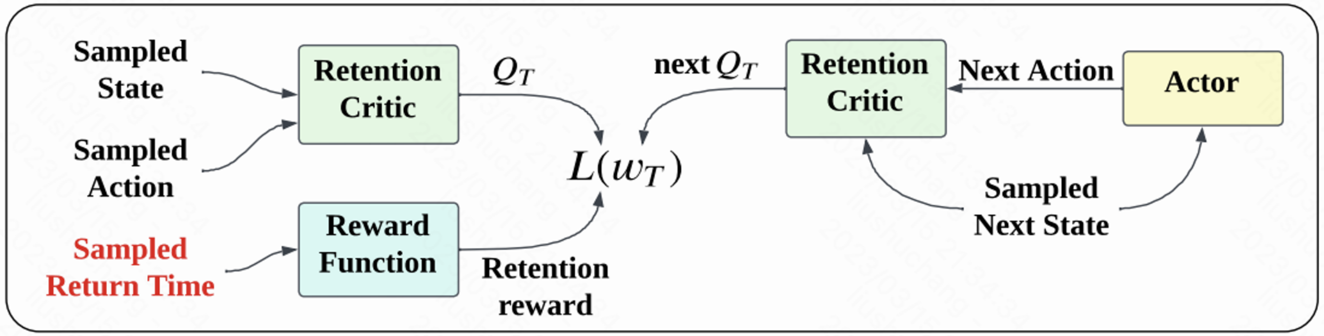
Reinforcement Learning for User Retention Algorithm



Learning the Retention and Tackling the Uncertainty Challenge

b) Inference of RLUR

$$\sum_{s_{i_t},a_{i_t}\in D}\left(Q_T(s_{i_t},a_{i_t}|w_T)-(r(s_{i_t},a_{i_t})+\gamma_{i_t}Q_T(s_{i_{t+1}},\pi(s_{i_{t+1}}|\theta)|w_T))\right)^2 \quad (1)$$



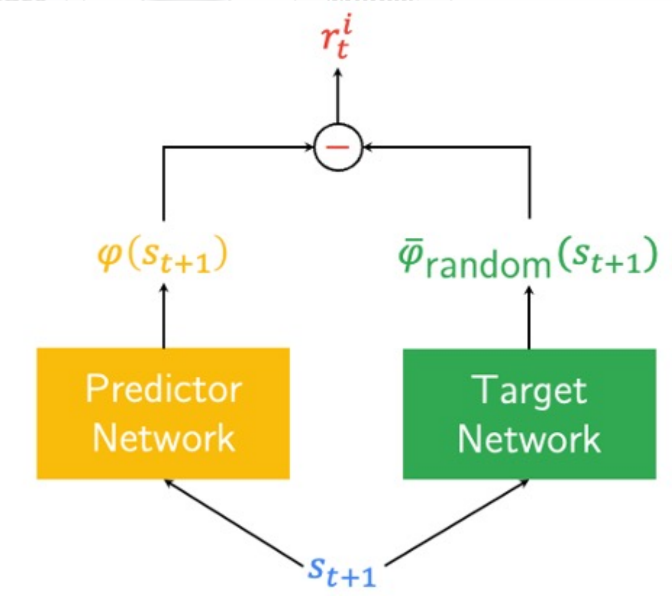
d) Retention critic learning of RLUR

- A normalization technique to reduce the variance
  - Learn a session level classification model $\mathrm{T}'(x)$
    - predict that the time is shorter than T_β
  - Estimate the lower bound of returning time by Markov Inequality
    - $(1 - T'(x)) * T_\beta$
  - Use true returning time/estimated returning time as the retention reward

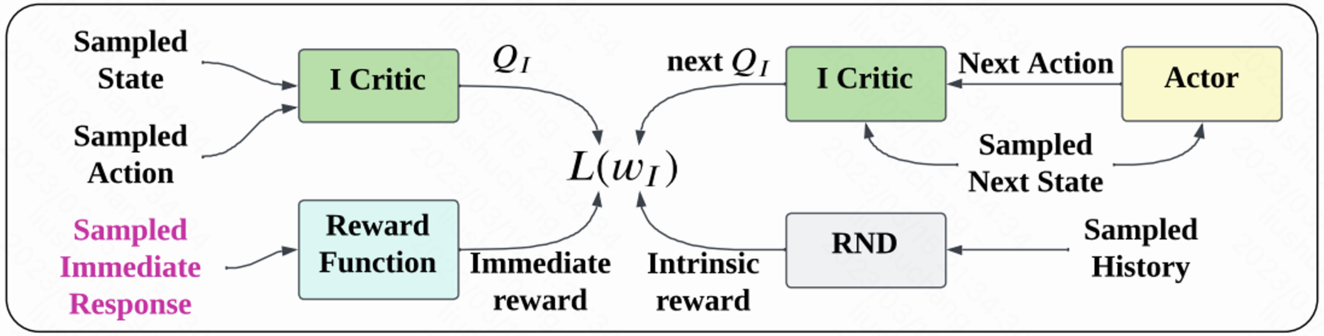
$$clip\{0, \frac{T(s_i)}{(1-T'(x))*T_\beta}, \alpha\}$$

Enhancing Learning by Heuristic Rewards

Intrinsic rewards by Random Network Distiliation



Enhancing policy learning with intrinsic rewards and immediate feedbacks



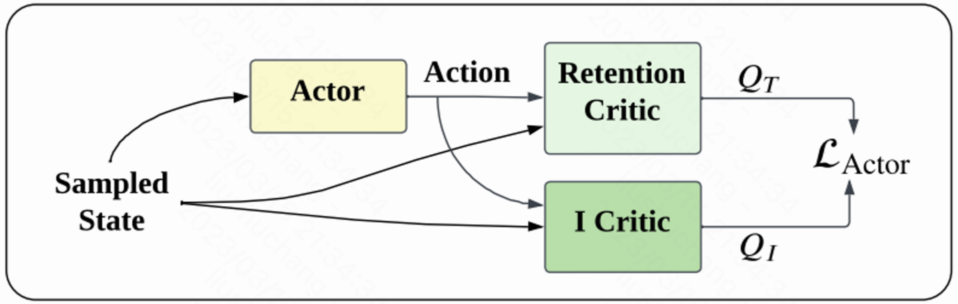
e) Immediate response critic learning of RLUR

Tackling the Unstable Training and Bias Problem

- Problem of previous regularization methods
  - $L(\theta) + \alpha KL(N(\pi_\theta(s), \delta), N(\mu, \delta))$
  - Learn either too slow or too fast

- Soft regularized actor loss: $\exp(\max\{\lambda * (log(p(a_{i_t}|s_{i_t})) - log(p_b(a_{i_t}|s_{i_t}))), 0\}) L(\theta)$
  - Samples with larger shift gets smaller weights
  - λ controls the regularization degree

Actor learn from both retention critic and immediate response critic. $L(\theta_{\text{high}}) = \lambda_T Q_T(s_{i_t}, \pi(s_{i_t}|\theta_{\text{high}})|w_T) - \lambda_I Q_I(s_{i_t}, \pi(s_{i_t}|\theta_{\text{high}})|w_I).$



c) Actor training of RLUR

Solving bias: different policies for users of different activity

Offline and Live Experiments

Table 1: Offline Results

Algorithm	Returning time↓	User retention↑
CEM	2.036	0.587
TD3	2.009	0.592
RLUR (naive, $\gamma = 0$)	2.001	0.596
RLUR (naive, $\gamma = 0.9$)	1.961	0.601
RLUR	**1.892**	**0.618**

- State
  - user profile
    - age, gender, and location
  - behavior history
    - user statistics, video id and user's feedback of in previous 3 requests
  - the candidate video features
- Action
  - 8-dimensional continuous vector ranging in $[0, 4]$
- Immediate Reward
  - sum of watch time and interactions of 6 videos

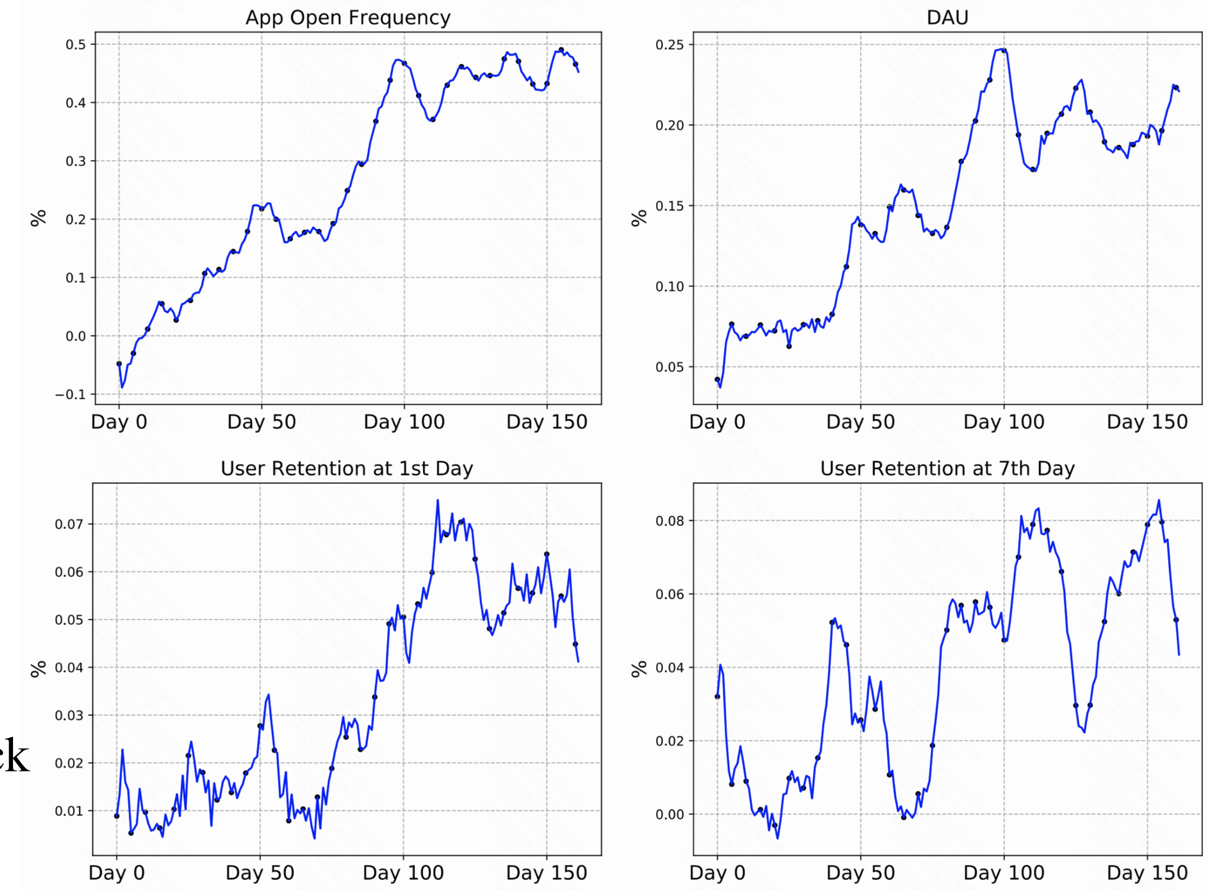


Figure 2: Live performance gap of each day.

Summary

- RL for Short Video RS
  - Hyparameter tuning and Ranking
  - Multi-objectives and delayed feedback
- Code Implementaions of our RL-based works
  - https://github.com/ksRecoTech/Wonderful-RL4Rec/tree/main

Long Paper

Cai, Qingpeng, et al. "Two-Stage Constrained Actor-Critic for Short Video Recommendation."
Proceedings of the ACM Web Conference 2023(WWW 2023). [code]
Keywords: multi-objective, main and auxiliary objectives, actor-critic

Liu, Shuchang, et al. "Exploration and Regularization of the Latent Action Space in Recommendation."
Proceedings of the ACM Web Conference 2023(WWW 2023). [code]
Keywords: latent action space, sequential recommendation, hyper-actor critic

Liu, Ziru, et al. "Multi-Task Recommendations with Reinforcement Learning."
Proceedings of the ACM Web Conference 2023(WWW 2023). [code]
Keywords: multi-task learning, xtr prediction

Xue, Wanqi, et al. "ResAct: Reinforcing Long-term Engagement in Sequential Recommendation with Residual Actor."
International Conference on Learning Representations(ICLR), 2023. [code]
Keywords: offline rl, sequential recommendation

Liu, Shuchang, et al. "Generative Flow Network for Listwise Recommendation."
Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining(KDD 2023). [code]
Keywords: generative model, list-wise recommendation

Xue, Wanqi, et al. "PrefRec: Recommender Systems with Human Preferences for Reinforcing Long-term User Engagement."
SIGKDD Conference on Knowledge Discovery and Data Mining(KDD 2023). [code]
Keywords: rlhf, preference modeling, sequential recommendation