Reinforce Lifelong Interaction Value of User-Author Pairs for Large-Scale Recommendation Systems

Yisha Li
Kuaishou Technology
Beijing, China
liyisha@kuaishou.com

Lexi Gao
Kuaishou Technology
Beijing, China
gaolexi@kuaishou.com

Jingxin Liu[*]
Kuaishou Technology
Beijing, China
liujingxin05@kuaishou.com

Xiang Gao
Kuaishou Technology
Beijing, China
gaoxiang12@kuaishou.com

Xin Li
Kuaishou Technology
Beijing, China
lixin05@kuaishou.com

Haiyang Lu
Kuaishou Technology
Beijing, China
luhaiyang@kuaishou.com

Liyin Hong
Kuaishou Technology
Beijing, China
hongliyin@kuaishou.com

ABSTRACT

Recommendation systems (RS) help users find interested content and connect authors with their target audience. Most research in RS tends to focus either on predicting users' immediate feedback (like click-through rate) accurately or improving users' long-term engagement. However, they ignore the influence for authors and the lifelong interaction value (LIV) of user-author pairs, which is particularly crucial for improving the prosperity of social community on different platforms. Currently, reinforcement learning (RL) can optimize long-term benefits and has been widely applied in RS. In this paper, we introduce RL to **R**einforce **L**ifelong **I**nteraction **V**alue of **U**ser-**A**uthor pairs (RLIV-UA) based on each interaction of UA pairs. To address the long intervals between UA interactions and the large scale of the UA space, we propose a novel Sparse Cross-Request Interaction Markov Decision Process (SCRI-MDP) and introduce an Adjacent State Approximation (ASA) method to construct RL training samples. Additionally, we introduce Multi-Task Critic Learning (MTCL) to capture the progressive nature of UA interactions (click → follow → gift), where denser interaction signals are leveraged to compensate for the learning of sparse labels. Finally, an auxiliary supervised learning task is designed to enhance the convergence of the RLIV-UA model. In offline experiments and online A/B tests, the RLIV-UA model achieves both higher user satisfaction and higher platform profits than compared methods.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → *Reinforcement learning*..

KEYWORDS

Recommendation System, Lifelong Interaction Value, Reinforcement Learning, Sparse Cross-Request Interaction Markov Decision Process, Multi-Task Critic Learning

ACM Reference Format:
Yisha Li, Lexi Gao, Jingxin Liu, Xiang Gao, Xin Li, Haiyang Lu, and Liyin Hong. 2025. Reinforce Lifelong Interaction Value of User-Author Pairs for Large-Scale Recommendation Systems. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The recommendation system (RS) aims to help users discover content aligned with their interests, while simultaneously enabling content authors to reach their target audiences, thereby facilitating fan accumulation and revenue generation [1, 23, 45]. By fostering repeated, mutually beneficial interactions between users and authors, RS plays a pivotal role in cultivating vibrant platform ecosystems, ultimately driving increased user engagement, traffic, and commercial returns [2, 10, 20–22, 29, 31].

Current research in RS largely falls into two categories. The first focuses on improving the accuracy of immediate user feedback prediction shown in Fig 1, such as click-through rate (CTR), at each recommendation request, typically using deep neural networks (DNNs) [8, 15, 18, 25, 50]. The second category leverages reinforcement learning (RL) to optimize long-term user engagement from **the user's perspective** [4, 7, 41, 42, 44, 47, 49, 51], dynamically maximizing session-level or trajectory-level cumulative rewards [11, 36, 40].

However, both paradigms largely overlook a critical dimension: **the Lifelong Interaction Value (LIV) of user-author (UA) pairs**.

[*] Corresponding author.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

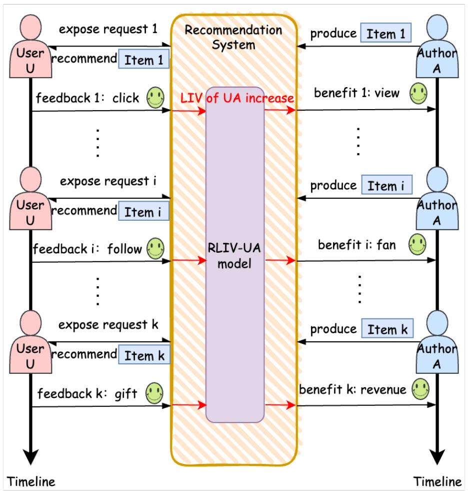


Figure 1: The main process of the proposed RLIV-UA model optimizing the LIV of UA pair based on their interactions.

By neglecting the bidirectional, evolving nature of UA relationships, existing methods fail to capture how sustained interactions, from initial discovery to deep loyalty, jointly benefit both parties and, by extension, the platform itself. This omission is particularly consequential, as the long-term stickiness and vitality of platform communities hinge on nurturing these relationships.

As further evidenced in Appendix C, we observe a strong positive correlation between the depth and frequency of lifelong UA interactions and key platform-level outcomes, including total revenue and average app usage time. This underscores a fundamental insight: strategically modeling and reinforcing the LIV of UA pairs is not merely a user- or author-centric optimization problem, but a direct pathway to maximizing overall platform prosperity.

To the best of our knowledge, this paper presents the first framework to **R**einforce **L**ifelong **I**nteraction **V**alue of **U**ser-**A**uthor pairs (RLIV-UA) using RL. As shown in Fig. 1, RLIV-UA dynamically optimizes the cumulative value of each UA interaction to progressively strengthen the mutual stickiness between user U and author A. Directly applying RL to model LIV, however, introduces significant challenges: (1) UA interactions often span long, irregular time intervals, unlike the dense, request-aligned interactions typically modeled in RS; (2) The combinatorial scale of the UA state space, driven by hundreds of millions of users and authors, renders it infeasible to store complete historical interaction traces over extended periods (e.g., months).

To address these challenges, we propose a novel **S**parse **C**ross-**R**equest **I**nteraction **M**arkov **D**ecision **P**rocess (SCRI-MDP) to formally model sparse, long-interval UA dynamics; an **A**djacent **S**tate **A**pproximation (ASA) method to construct practical RL training samples under storage constraints; and a **M**ulti-**T**ask **C**ritic **L**earning (MTCL) architecture [28] that captures the progressive nature of UA relationships (e.g., click → follow → gift), leveraging denser signals to bootstrap learning of sparse, high-value actions. Finally, to combat sample sparsity and label variance, we introduce a supervised learning task to stabilize training and accelerate convergence.

In summary, the key contributions of this work are:

- We propose RLIV-UA, a novel reinforcement learning framework designed to optimize the lifelong interaction value of user-author pairs in large-scale recommendation systems.
- We introduce SCRI-MDP, a formalism for modeling sparse, cross-request UA interactions, along with ASA, a practical method for constructing RL training samples under industrial constraints.
- We design an MTCL architecture to model the hierarchical progression of UA relationships (e.g., from clicks to gifts), augmented by an auxiliary supervised learning task to ensure stable and efficient training.
- Extensive offline simulations and large-scale online A/B tests demonstrate that RLIV-UA significantly improves both user engagement and author benefits, leading to substantial gains in platform revenue.

2 PROBLEM FORMULATION

Existing RL methods in RS often model user behaviors as infinite request-level markov decision process (MDP) [36]. Specifically, the time interval between adjacent states $\Delta = 1$ always holds. However, under the UA interaction space, the time interval between the same UA pair's adjacent interactions satisfies $\Delta \geq 1$. For a specific UA pair, the interactions are usually sparse due to the large scale of candidate recommended items. Therefore, we define a novel sparse cross-request interaction MDP to model the LIV of UA pairs:

Definition 2.1 (Sparse Cross-Request Interaction MDP). A *Sparse Cross-Request Interaction MDP* (SCRI-MDP) is a tuple represented as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \Delta \rangle$, where:

- **State space** $\mathcal{S}$: Each state $\mathbf{s}^t_{\text{ua}} \in \mathcal{S}$ encodes user features $\mathbf{f}^t_{\text{u}}$, author features $\mathbf{f}^t_{\text{a}}$ and interaction features $\mathbf{f}^t_{\text{ua}}$ (e.g., cumulative watch time, gift count).
- **Action space** $\mathcal{A}$: The action $a^t_{\text{ua}} \in \mathcal{A}$ is the weight of the ranking score at request t.

$$final\_score = rank\_score * (c + a^t_{\text{ua}} * w)^b \tag{1}$$

  where w, b, c are hype-parameters.
- **State transition distribution** $\mathcal{P}$: $P(\mathbf{s}^{t+1}_{\text{ua}} \mid \mathbf{s}^t_{\text{ua}}, a^t_{\text{ua}}, \Delta^t_{\text{ua}})$ is determined by the time gap $\Delta^t_{\text{ua}} \geq 1$ between consecutive interactions of the same UA pair, skipping intermediate requests for that pair if no interaction occurs.
- **Reward function** $\mathcal{R}$: To model the progressive changes of UA relationship, several functions are designed to model the long-term reward of different immediate feedback between UA. Let $r_{\text{ua},c}, r_{\text{ua},w}, r_{\text{ua},f}, r_{\text{ua},g}$ be the reward function of click, effective view, follow, and gift labels, respectively. $C = \{c, w, f, g\}$ denotes the target label set and $n = 4$ denotes the cardinality of C.

$$r_{\text{ua},l} = \begin{cases} 1, & \text{the behavior } l \text{ happens}, l \in C \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

- **Discount factor** $\gamma \in [0, 1)$: Balances immediate and future rewards.

- **Interaction gap** Δ^t_{ua}: Number of global requests between two consecutive interactions of the same UA pair.

Note that the proposed SCRI-MDP is a specific instantiation of the general semi-Markov Decision Process (semi-MDP) framework [37]. In a standard semi-MDP, actions can take variable amounts of time to complete, and state transitions occur upon the completion of these 'macro-actions'. In our case, the 'macro-action' corresponds to the recommendation policy applied between two consecutive interactions of a UA pair. The decision point (state transition) only occurs when an actual UA interaction happens, which naturally fits the semi-MDP paradigm by handling the variable and often long time intervals ($\Delta \geq 1$) between interactions. Furthermore, the proposed SCRI-MDP can be regarded as an augmented MDP to convert the semi-MDP paradigm into a 'delay-free' standard MDP paradigm. Then we introduces the following theorem.

Theorem 2.2 (Equivalence to augmented MDP). *Consider a Sparse Cross-Request Interaction MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \Delta \rangle$ with interaction gap variable $\Delta^t_{\text{ua}} \geq 1$. Define an augmented state $\bar{s}_t = (s^t_{\text{ua}}, \Delta^t_{\text{ua}})$ where Δ^t_{ua} is treated as part of the state vector, forming an augmented state space $\bar{\mathcal{S}} = \mathcal{S} \times \mathbb{N}$. Then $\mathcal{M}$ is equivalent to a standard MDP $\bar{\mathcal{M}} = \langle \bar{\mathcal{S}}, \mathcal{A}, \bar{\mathcal{P}}, \mathcal{R}, \gamma \rangle$, where the transition kernel is defined as:*

$$\bar{P}((\mathbf{s}', \Delta') \mid (\mathbf{s}, \Delta), a) = P(\mathbf{s}'_{\text{ua}}, \Delta'_{\text{ua}} \mid \mathbf{s}_{\text{ua}}, \Delta_{\text{ua}}, a_{\text{ua}}), \tag{3}$$

and the reward function $\mathcal{R}$ remains identical. Any optimal policy π^* in $\mathcal{M}$ corresponds to an optimal policy in $\bar{\mathcal{M}}$ under this mapping.

The proof of 2.2 is available in the Appendix A.1. Theorem 1 establishes a formal equivalence between the proposed SCRI-MDP and a standard MDP. It provides a rigorous justification for applying conventional MDP-based reinforcement learning algorithms (such as DQN, DDPG, or TD3, which are grounded in standard MDP theory) to solve the SCRI-MDP problem. By demonstrating that the SCRI-MDP can be transformed into an equivalent standard MDP through state augmentation, we ensure that the theoretical convergence properties and optimality guarantees of these well-studied algorithms remain applicable to our framework. This bridges the gap between our novel, problem-specific formulation and the vast body of existing RL theory and practice.

Based on the aforementioned SCRI-MDP formulation, we introduce a multi-task critic learning (MTCL) architecture to model the LIV of UA pairs by capturing the progressive changes of lifelong UA relationship. There are n critic networks $Q_{\phi_k}, k = 1, ..., n$ with ϕ_k as their trainable parameters to optimize the corresponding cumulative rewards $r_{\text{ua},c}, r_{\text{ua},w}, r_{\text{ua},f}, r_{\text{ua},g}$. The final objective function is as follows:

$$\max_{\phi_1, \cdots, \phi_n} \sum_{l \in C} \mathbb{E}\left[\sum_{t=0}^{\infty} r^t_{\text{ua},\,l}\right] \tag{4}$$

Each critic network is designed to learn from the states of UA pairs and corresponding rewards. Specifically, at one request of user u, RS recommends author a's item by outputting the action $a^t_{\text{ua}} \in \mathcal{A}$ which is the t-th interaction of ua, then the optimal LIV of ua pair and action a^t_{ua} is defined as:

$$Q^*(\mathbf{s}^t_{\text{ua}}, a^t_{\text{ua}}) = r^t_{\text{ua}} + \gamma \sum_{\mathbf{s}^{t+1}_{\text{ua}} \in \mathcal{S}} P(\mathbf{s}^{t+1}_{\text{ua}} \mid \mathbf{s}^t_{\text{ua}}, a^t_{\text{ua}}) \max_{a^{t+1}_{\text{ua}} \in \mathcal{A}} Q^*(\mathbf{s}^{t+1}_{\text{ua}}, a^{t+1}_{\text{ua}}) \tag{5}$$

3 METHODOLOGY

In this section, we propose the RLIV-UA model to represent the SCRI-MDP in the UA pair state space. Firstly, due to the long time span between adjacent states of UA pair in the SCRI-MDP, we propose the Adjacent State Approximation (ASA) method to **construct RL training samples**. Then, we introduce the detailed network architecture of multi-task LIV networks and the final online deployment of the RLIV-UA model. The algorithm pseudocode of the RLIV-UA model is presented in Appendix B.3.

3.1 Adjacent State Approximation

At the t-th interaction of a UA pair, the next state $\mathbf{s}^{t+1}_{\text{ua}}$ is delayed until the next interaction which may occur much later. Thus, the traditional RL training sample $(\mathbf{s}^t_{\text{ua}}, a^t_{\text{ua}}, r^t_{\text{ua}}, \mathbf{s}^{t+1}_{\text{ua}})$ cannot be formed in real time due to sparse, infrequent interactions. In industrial-scale recommendation systems, the sheer number of users and authors makes it infeasible to store the state of every UA pair in a key-value database. Instead, systems typically store user behavior as fixed-length sequential logs. Due to finite storage capacity, these sequences are truncated, which often prevents the system from retrieving a UA pair's prior interaction, especially for infrequent or new relationships. In our online deployment, the success rate of constructing consecutive RL training samples $(\mathbf{s}^{t-1}_{\text{ua}}, a^{t-1}_{\text{ua}}, r^{t-1}_{\text{ua}}, \mathbf{s}^t_{\text{ua}})$ by stitching historical logs is only 47%. Training LIV networks directly on such sparse and fragmented samples would lead to severe model bias, the RL agent learns from an incomplete, non-representative subset of UA trajectories, significantly degrading its ability to optimize long-term, bidirectional value. Moreover, learning from stitched historical tuples inherently delays the model's ability to act on new relationships. Specifically, the very first interaction between a user and an author cannot be used for training until a **second** interaction occurs. This creates a critical blind spot: the model is unable to learn how to nurture the **initiation phase** of a UA relationship.

To address these challenges, we propose the **A**djacent **S**tate **A**pproximation (ASA) mechanism, a parameterized state predictor f_θ to reconstruct the missing next state from the current interaction:

$$\hat{\mathbf{s}}^{t+1}_{\text{ua}} = f_\theta(\mathbf{s}^t_{\text{ua}}, a^t_{\text{ua}}, r^t_{\text{ua}}) \tag{6}$$

where the state $\mathbf{s}^t_{\text{ua}} = \{\mathbf{f}^t_{\text{u}}, \mathbf{f}^t_{\text{a}}, \mathbf{f}^t_{\text{ua}}, \Delta^t_{\text{ua}}\}$ is augmented with the interaction gap Δ^t_{ua}, and θ denotes the trainable parameters of f_θ.

As illustrated in Fig. 2, we first train f_θ by retrieving the most recent interaction between the same UA pair from the user's historical behavior logs. Industrial systems typically store interaction events (e.g., clicks, follows, gifts) as fixed-length sequences. To construct a supervised training sample, we first locate the timestamp T of the user's most recent interaction with author a. Since watching an item to completion typically marks the end of an interaction episode, we perform a binary search within a temporal window around T to retrieve the corresponding action a^{t-1}_{ua} and associated feedback signals. This forms the training tuple $(\mathbf{s}^{t-1}_{\text{ua}}, a^{t-1}_{\text{ua}}, r^{t-1}_{\text{ua}}, \mathbf{s}^t_{\text{ua}})$, optimized via mean squared error:

$$\mathcal{L}_\theta = \left\| f_\theta(\mathbf{s}^{t-1}_{\text{ua}}, a^{t-1}_{\text{ua}}, r^{t-1}_{\text{ua}}) - \mathbf{s}^t_{\text{ua}} \right\|_2^2 \tag{7}$$

With ASA, then we can approximate the full RL training sample $(\mathbf{s}^t_{\text{ua}}, a^t_{\text{ua}}, r^t_{\text{ua}}, \hat{\mathbf{s}}^{t+1}_{\text{ua}})$ for **every** interaction shown in Fig. 2, not just

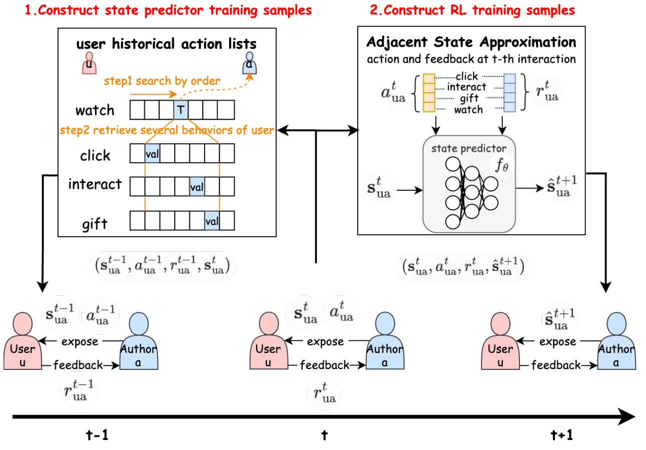

1.Construct state predictor training samples
2.Construct RL training samples
user historical action lists
u
a
step1 search by order
watch
T
step2 retrieve several behaviors of user
click
val
interact
val
gift
val
Adjacent State Approximation
action and feedback at t-th interaction
a^t_{ua}
click
interact
gift
watch
r^t_{ua}
state predictor
f_θ
$\mathbf{s}^t_{ua}$
$\hat{\mathbf{s}}^{t+1}_{ua}$
$(\mathbf{s}^{t-1}_{ua}, a^{t-1}_{ua}, r^{t-1}_{ua}, \mathbf{s}^t_{ua})$
$(\mathbf{s}^t_{ua}, a^t_{ua}, r^t_{ua}, \hat{\mathbf{s}}^{t+1}_{ua})$
$\mathbf{s}^{t-1}_{ua}$ a^{t-1}_{ua}
User u
expose
feedback
Author a
r^{t-1}_{ua}
$\mathbf{s}^t_{ua}$ a^t_{ua}
r^t_{ua}
$\hat{\mathbf{s}}^{t+1}_{ua}$
t-1
t
t+1

Figure 2: Training sample construction for state predictor and RL model.

those with retrievable history. This directly addresses the 53% stitching failure rate and eliminates the cold-start blind spot for new UA pairs. Crucially, ASA enables the RL agent to learn from **first-time interactions**, capturing the full lifecycle of UA relationships, from initial discovery to deep, monetizable engagement, and ensuring comprehensive learning across the entire interaction space.

3.2 Theoretical Analysis

The proposed adjacent state approximation enables practical training of RL models in industrial recommendation systems. However, a critical question arises: *How does the approximation error in ASA affect the optimality of the learned policy?* To address this, we establish the following theorem that provides an error bound on the Q-function when using approximate state transitions.

Theorem 3.1 (Error Bound for Approximate State Transitions). *Let Q^* be the optimal Q-function of the SCRI-MDP $\mathcal{M}$, and let $\hat{Q}$ be the Q-function learned using the ASA with maximum prediction error $\epsilon = \max_{s,a,r} \left|\hat{s}^{t+1} - s^{t+1}\right|$. If the ASA error ϵ is bounded and the reward function is Lipschitz continuous with constant L_r, then the difference between $\hat{Q}$ and Q^* is bounded by:*

$$\left|\hat{Q}(s,a) - Q^*(s,a)\right| \leq \frac{L_r \epsilon}{(1-\gamma)^2} \tag{8}$$

for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, where $\gamma \in [0,1)$ is the discount factor.

The proof of it is available in the Appendix A.2. This theorem provides a theoretical guarantee that the performance degradation caused by ASA is bounded and decreases as the ASA error ϵ decreases. In practice, we can train the ASA network f_θ to minimize ϵ, ensuring that the learned policy remains close to optimal. Note that the Lipschitz continuity of the reward function is a standard and widely adopted assumption in theoretical RL literature to ensure the stability and boundedness of value functions. In our implementation, the reward functions are designed as bounded, non-negative functions of user engagement. This design choice inherently promotes smoothness in the reward landscape, making the Lipschitz assumption practically reasonable for our problem setting.

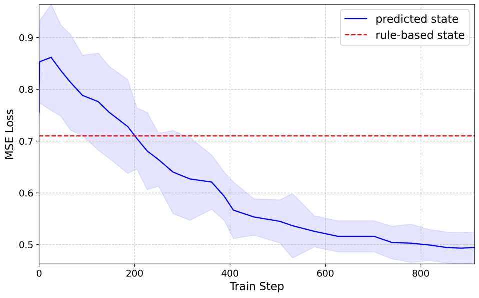


Figure 3: The training process of ASA.

We further validate the effectiveness of ASA by comparing it against a rule-based baseline, as illustrated in Fig. 3. The ASA method achieves a low MSE loss during training, which directly implies a small approximation error ϵ in estimating the next state representation $\mathbf{s}^{t+1}$. It further implies that the RL samples generated by ASA are of high fidelity. Consequently, when RLIV-UA is trained on these samples, it provably converges to a near-optimal policy, which satisfies the Q-value performance bound established in Theorem 3.1. More discussion and offline evaluation results of ASA is detailed in Appendix D.

Overall, ASA is a storage-constrained approximation mechanism designed for industrial-scale recommendation systems, where storing complete historical traces for all user-author pairs is infeasible. Unlike model-based RL approaches, e.g. Dreamer [16], that learn transition dynamics for planning or sample efficiency, ASA serves a singular, pragmatic purpose: to reconstruct valid (s, a, r, s') tuples for RL training when the true next state s' is physically unavailable due to system limitations. Its objective is not to predict environmental dynamics, but to preserve training signal continuity under extreme data sparsity.

3.3 Multi-Task LIV Networks

The overall framework of the RLIV-UA model is illustrated in Fig. 4. To capture the progressive nature of lifelong user-author relationships, from initial discovery to deep engagement, we design four dedicated LIV networks based on the Multi-Task Critic Learning (MTCL) architecture: Click, Effective View, Follow, and Gift. Crucially, the positive sample rates for these signals vary drastically across tasks: approximately 3% for Click, 1% for Effective View, 0.2% for Follow, and a mere 0.01% for Gift. This extreme imbalance renders the Follow and Gift labels exceptionally sparse, making direct learning from them highly unstable and inefficient. To address this, MTCL leverages the relatively dense signals (Click and Effective View) as auxiliary tasks to bootstrap and stabilize the learning of the sparse, high-value actions (Follow and Gift), enabling the model to effectively propagate reward signals across the entire interaction hierarchy. For simplicity, we use a single task tower as an example to illustrate the network structure below.

As shown in Fig. 4, the current UA state $\{\mathbf{f}^t_{\mathrm{u}}, \mathbf{f}^t_{\mathrm{a}}, \mathbf{f}^t_{\mathrm{ua}}, \Delta^t_{\mathrm{ua}}\}$ is fed into a shared embedding layer to obtain corresponding hidden embeddings $\mathbf{h}^t_{\mathrm{u}}$, $\mathbf{h}^t_{\mathrm{a}}$, $\mathbf{h}^t_{\mathrm{ua}}$ and δ^t_{ua}. Taking the vector $\mathbf{H}^t_{\mathrm{ua}} = \mathrm{concat}(\mathbf{h}^t_{\mathrm{u}}, \mathbf{h}^t_{\mathrm{a}}, \mathbf{h}^t_{\mathrm{ua}}, \delta^t_{\mathrm{ua}})$ as the network input, $Q_\phi(\mathbf{H}^t_{\mathrm{ua}}, a^t_{\mathrm{ua}})$ is denoted as the final LIV of ua at the t-th interaction. To mitigate the

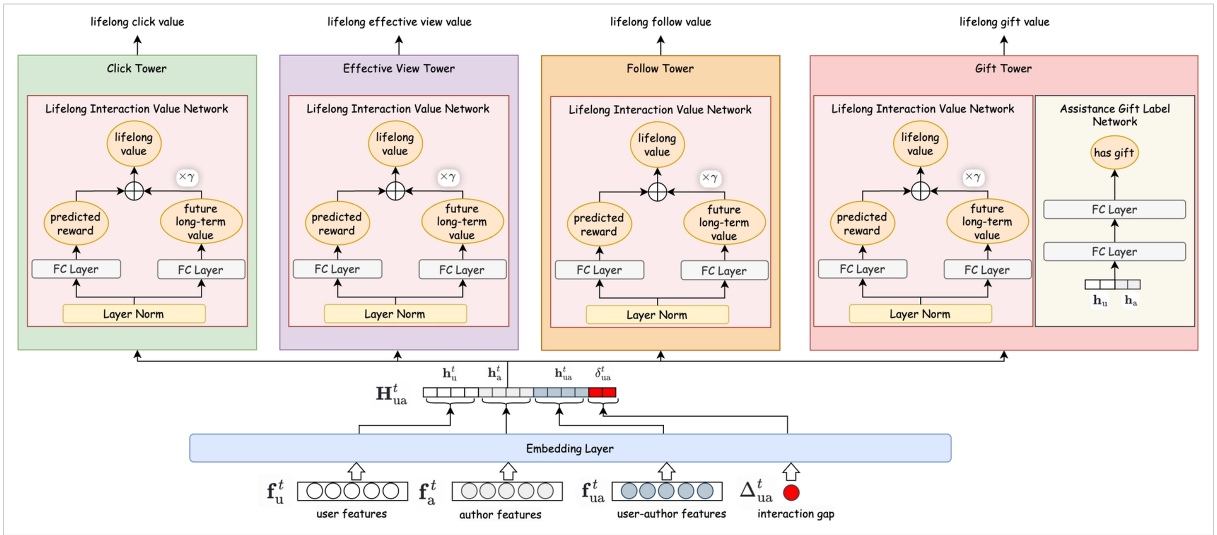


Figure 4: The overall framework of the proposed RLIV-UA model.

problem of value overestimation deviation [40], double value networks and two corresponding target networks are used to generate the minimum value:

$$Q_\phi(\mathbf{H}^t_{ua}, a^t_{ua}) = min(Q_{\phi^1}(\mathbf{H}^t_{ua}, a^t_{ua}), Q_{\phi^2}(\mathbf{H}^t_{ua}, a^t_{ua})) \qquad (9)$$

Then the corresponding loss function of a LIV network is defined as follows:

$$\mathcal{L}(\phi) = \mathbb{E}_{(\mathbf{s}^t_{ua}, a^t_{ua}, r^t_{ua}, \hat{\mathbf{s}}^{t+1}_{ua})\in D}[(Q_\phi(\mathbf{H}^t_{ua}, a^t_{ua}) - y)^2] \qquad (10)$$

$$y = r^t_{ua} + \gamma Q'_{\phi'}(\mathbf{H}^{t+1}_{ua}, \operatorname*{argmax}_{a^{t+1}_{ua}\in\mathcal{A}} Q_\phi(\mathbf{H}^{t+1}_{ua}, a^{t+1}_{ua})) \qquad (11)$$

where D indicates the sample buffer collected in real time, y indicates the target output value of the LIV network, and $Q'_{\phi'}$ represents the output value of the target network with the same structure as the LIV network Q_ϕ. Note that the network parameter ϕ' of $Q'_{\phi'}$ is periodically copied from ϕ of Q_ϕ.

To reinforce the learning of the sparse gift label of UA pairs in the Gift tower, an assistance MLP network is designed to predict whether user U will gift author A at this interaction. As shown in Fig. 4, the user and the author static hidden embeddings $\mathbf{h}_u, \mathbf{h}_a$ are input to the assistance network. The binary cross-entropy loss $\mathcal{L}^g_A$ of the assistance gift binary classification goal is added to the total loss.

$$\mathcal{L}^g_A = -\mathbb{E}_{\mathbf{s}^t_{ua}\in D}\left[y_g \log(\hat{f}(\mathbf{h}_u, \mathbf{h}_a)) + (1-y_g)\log(1-\hat{f}(\mathbf{h}_u, \mathbf{h}_a))\right] \qquad (12)$$

where y_g indicates the true value of whether user U will gift author A at this interaction and $\hat{f}(\mathbf{h}_u, \mathbf{h}_a)$ is the predicted value output by assistance network.

3.4 Auxiliary Supervised Learning Network

Previous work [26] finds that the target value y is often dominated by the inaccurate output of the target network $Q'_{\phi'}$ in practice, due to the instability of critic learning in RL. This problem reduces the effectiveness of the real reward r^t_{ua} in guiding the learning of the value network, since it becomes relatively too small to provide meaningful learning signals. Furthermore, the large scale and extreme sparsity of the UA state space make the RL model even more difficult to converge.

Therefore, we introduce an auxiliary supervised learning network to regulate the learning of each LIV network Q_ϕ, preventing a potential divergence of the RL model. Specifically, each LIV network is divided into two parts as follows:

$$Q_\phi(\mathbf{H}^t_{ua}, a^t_{ua}) := \hat{R}_\eta(\mathbf{H}^t_{ua}, a^t_{ua}) + \gamma \times \hat{T}_\xi(\mathbf{H}^t_{ua}, a^t_{ua}) \qquad (13)$$

where $\hat{R}_\eta$ represents the reward prediction network with η as its trainable parameters, $\hat{T}_\xi$ is the Q residual network with ξ as its trainable parameters and $\phi = \{\eta, \xi\}$.

Since the real reward r^t_{ua} is available based on the t-th interaction between ua, the predicted reward $\hat{R}_\eta(\mathbf{H}^t_{ua}, a^t_{ua})$ can be learned by supervised loss. Incorporating with the aforementioned clipped double Q-learning [12] shown in Eq. 9, the auxiliary supervised learning network improves the convergence of the RLIV-UA model.

Hence, the general loss of an LIV network is defined as

$$\mathcal{L}_Q = MSE(\hat{R}_\eta(\mathbf{H}^t_{ua}, a^t_{ua}), r^t_{ua}) + \sum_{k=1}^{2} MSE(Q_{\phi^k}(\mathbf{H}^t_{ua}, a^t_{ua}), y) \qquad (14)$$

where the first term denotes the loss for the auxiliary supervised learning network and the second term denotes the original critic learning loss which stops gradients for $\hat{R}_\eta$.

Overall, for the whole multi-task LIV networks, the final loss function is defined as follows:

$$\mathcal{L} = \sum_{l\in C} \mathcal{L}^l_Q + \mathcal{L}^g_A \qquad (15)$$

where $\mathcal{L}^g_A$ is the assistance gift loss (defined in Eq. 12) and $\mathcal{L}^l_Q$ is the loss function (defined in Eq. 14), for each label in the target label set C, respectively.

3.5 Online Deployment

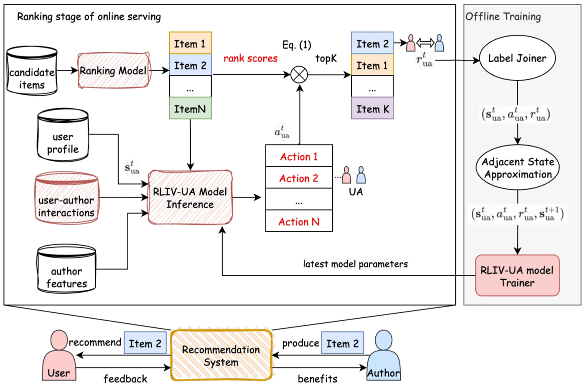


Figure 5: System architecture of the RLIV-UA model in real industrial recommendation scenario.

In industrial RS, the online system architecture of RLIV-UA model is shown in Fig. 5, including the offline training process and the online serving process. In the offline training process, we first utilize the label joiner to merge the UA state features $\mathbf{s}^t_{\text{ua}}$, the action a^t_{ua}, and the immediate feedback r^t_{ua} at timestamp t. Then we leverage the ASA method to approximate the next state $\mathbf{s}^{t+1}_{\text{ua}}$ and obtain RL training sample $(\mathbf{s}^t_{\text{ua}}, a^t_{\text{ua}}, r^t_{\text{ua}}, \hat{\mathbf{s}}^{t+1}_{\text{ua}})$. In the online serving process, the RLIV-UA model is deployed as a weight of rank score during the ranking stage of RS to influence the final ordering of candidate items, whose number N is typically less than 300.

Specifically, at each request of a user, the RLIV-UA model outputs actions with highest LIV value from different task towers for candidate items. For simplicity, there only shows the application process of the action corresponding to a certain tower in Fig 5. The candidate items are then ranked by Eq. 2.1 based on their final scores. Finally, the top K candidate items are selected as the final item list and are exposed to the user by order. Then the user interacts with each exposure item to return feedback signals to RS.

In practice, the action a^t_{ua} with highest LIV value from different task towers can be selectively applied based on the actual optimization goal, such as improving long-term user retention or maximizing platform revenue. To explore different actions for ranking, the ϵ-greedy strategy is applied online.

4 EXPERIMENTS

The RLIV-UA model is evaluated in an offline simulated recommendation scenario to compare its performance with state-of-the-art models and RLIV-UA variants. It is also applied in two real industrial recommendation scenarios to verify its effectiveness in large-scale industrial RS through online A/B tests.

4.1 Experimental Setup

4.1.1 Dataset and Evaluation Metrics. We adopt the KuaiRand dataset [13] as the foundation for our offline experiments. This public dataset, collected from the Kuaishou app, contains interactions from 27,285 users across 32,038,725 items, resulting in hundreds of millions of UA interaction records. We train an offline user simulator on this dataset to serve as a controllable environment that mimics real user behavior. Specifically, upon receiving a recommended item, the simulator generates immediate feedback signals, such as clicks, watch duration, and comments, and subsequently decides whether to issue the next request based on a probabilistic quitting mechanism, similar to the approach described in [46].

To comprehensively evaluate model performance, we assess three key dimensions: **user satisfaction**, **author benefits**, and **platform profitability**. The detailed illustration of all evaluation metrics is available in Appendix B.1.

- **User Satisfaction** is measured by: Session Length (number of requests per session), Watch Time (total viewing duration per session), and CTR (click-through rate).
- **Author Benefits** are quantified via: Diversity (variety of recommended authors/content) and New Fans (number of new followers acquired by authors).
- **Platform Profits** are evaluated using: UA Count (number of UA pairs exhibiting deep relationships), Weekly Gifted Users (number of users sending virtual gifts per week), Total Revenue, App Usage Time, and Weekly Retention (proportion of users returning weekly).

4.1.2 Compared Methods. We compare our method against a suite of representative baselines, including the classic **RankingModel** and four established reinforcement learning approaches: **CQL** [24], **DQN** [40], **TD3** [11], **TD3-UA** (Using TD3 to model the SCRI-MDP and the action is a continuous weight value), **FeedRec** [51] and **RLUR** [4]. To further validate the contribution of each component in RLIV-UA, we also evaluate four ablated variants: **RLIV-UA(w/o MT)** (without Multi-Task learning), **RLIV-UA(w/o MT & SL)** (without Multi-Task and Supervised Learning), **RLIV-UA(w/o AL & SL)** (without Auxiliary Gift Label Network and Supervised Learning), and **RLIV-UA(w/o SL)** (without Supervised Learning). The implementation details is available in Appendix B.2.

Note that the assistance gift label network is not applied in offline experiments, because there is no gift signals in KuaiRand dataset. The platform profits metrics are only used in online A/B experiments. All models are trained to convergence, and their results are the averaged performance of the last 10 epochs. Moreover, we use the follow LIV and gift LIV in Kwai and use the watch time LIV in Kuaishou and offline experiments.

4.2 Performance Comparison

The overall performance of different models in offline experiment is shown in Table 1. The traditional RankingModel achieves the best performance in CTR since it can predict which item has the greatest probability to be clicked. However, it is not suitable for improving the long-term user engagement such as session length and watch time. The offline model CQL learned from the historical samples can achieve some diversity. Compared with traditional RankingModel and the offline RL model CQL, most RL-based models achieve better performance in long-term metric session length at the expense of immediate feedback like low CTR, resulting in similar watch time. By adding another Q network learning from heuristic rewards, the RLUR model can improve all the long-term metrics including session length and watch time. The proposed RLIV-UA

Table 1: Overall performance of all compared models in offline recommendation scenario.

Models	Session Length	Watch Time	CTR	Diversity
RankingModel	2.0132	59.4812	**0.5948**	0.0629
CQL	2.2660	58.0753	0.5125	0.1727
DQN	5.3744	125.8436	0.4074	0.5801
TD3	4.2519	79.0432	0.4258	0.4929
TD3-UA	4.9651	102.0916	0.4465	0.7161
FeedRec	6.7420	118.8468	0.4363	0.7059
RLUR	6.6810	151.0550	0.4519	0.7206
RLIV-UA(w/o MT & SL)	7.2940	188.0499	0.5229	0.7994
RLIV-UA(w/o MT)	9.2800	248.1804	**0.5340**	0.8746
RLIV-UA	**12.8860**	**377.4867**	0.5015	**0.8827**

model achieves the best performance in session length and watch time and achieves the third high value in CTR, which reflects that the RLIV-UA model can balance immediate feedback and long-term feedback to improve long-term user engagement by modeling the LIV of UA pairs. Moreover, the RLIV-UA model achieves the best performance in diversity which reflects that modeling the LIV of UA pairs can more accurately recommend items of different authors to target users, rather than blindly recommend different items.

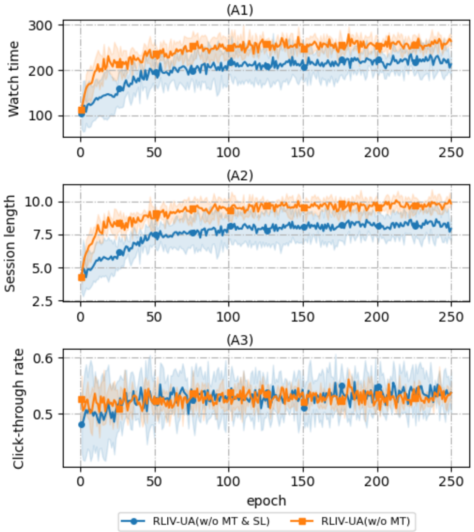


Figure 6: The learning process of RLIV-UA(w/o MT & SL) and RLIV-UA(w/o MT) over 10 rounds of training where the shaded areas correspond to the standard deviations.

4.3 Ablation Study

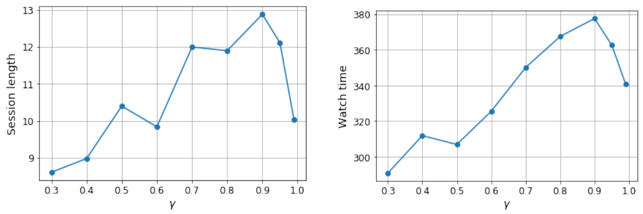
The overall performance of RLIV-UA variations in offline recommendation scenario are shown in Table 1.

Firstly, compared with the RLIV-UA(w/o MT & SL) variation, the RLIV-UA(w/o MT) achieves relatively high improvement in session length, watch time and diversity, which reflects that the auxiliary supervised learning task can help model learn the LIV of UA pairs more accurately.

Furthermore, the RLIV-UA achieves the best performance under both session length, watch time and diversity metrics, which indicates the effectiveness of the multi-task critic learning architecture.

As shown in Fig. 6, with the auxiliary supervised learning task, the variance of RLIV-UA(w/o MT) is much lower than that of RLIV-UA(w/o MT & SL) under watch time, session length and CTR metrics. It demonstrates the learning process of RLIV-UA(w/o MT) model is more stable, and the auxiliary supervised learning task is effective for enhancing the stability of model training.

As shown in Figure 7, the RLIV-UA model with $\gamma = 0.9$ achieves the best performance in session length and watch time and all the models achieve better performance compared with baselines, which shows the RLIV-UA model is not sensitive to the parameter γ.



(a) The session length of RLIV-UA with different γ.

(b) The watch time of RLIV-UA with different γ.

Figure 7: The parameter sensitivity experiment of γ.

4.4 Online A/B Experiments

The proposed RLIV-UA model and compared methods were deployed on the Kwai live-streaming feed, a platform serving over 100 million users and 1 million content authors, to optimize platform revenue from July to October 2024. For rigorous evaluation, we randomly assigned 20% of users (over 20 million) to the treatment group, with the remaining 80% serving as the control. All reported confidence intervals (CIs) in Table 2 are 95% two-sided CIs, computed based on the Central Limit Theorem, ensuring statistical robustness given the massive sample size.

As shown in Table 2, the performance of RLIV-UA and its ablated variants improves progressively: RLIV-UA(w/o AL&SL) → RLIV-UA(w/o SL) → full RLIV-UA. This consistent uplift across key business metrics, including Total Revenue +38.64%, Weekly Gifted Users

Table 2: Results of the revenue experiment on Kwai live-stream feed.

Models	UA Count	Weekly Gifted Users	Total Revenue	New Fans
RankingModel	-	-	-	-
RLIV-UA(w/o AL & SL)	+2.00% CI:[0.97%, 5.08%]	+2.25% CI:[0.46%, 4.08%]	+11.90% CI:[5.48%, 32.32%]	+2.74% CI:[0.60%, 4.92%]
RLIV-UA(w/o SL)	+3.58% CI:[1.18%, 5.36%]	+3.42% CI:[0.23%, 5.11%]	+23.81% CI:[15.93%, 37.52%]	+6.10% CI:[0.10%, 10.15%]
RLIV-UA(w/o ASA)	+2.02% CI:[1.71%, 5.74%]	+1.68% CI:[0.26%, 3.10%]	+15.21% CI:[0.36%, 21.34%]	+5.22% CI:[0.36%, 10.08%]
RLIV-UA	**+5.85% CI:[1.34%, 8.93%]**	**+5.37% CI:[0.11%, 8.71%]**	**+38.64% CI:[13.43%, 60.31%]**	**+8.28 CI:[0.91%, 14.38%]**

+5.37%, UA Count +5.85%, and New Fans +8.28%, demonstrates the practical effectiveness and additive value of each component in our framework. Notably, we also evaluate a variant, RLIV-UA(w/o ASA), which forgoes the ASA module and instead constructs RL training samples using only stitched historical interactions. This variant underperforms the full RLIV-UA model across all metrics, highlighting ASA's critical role in mitigating sample sparsity and bias. Without ASA, the model cannot learn from first-time or infrequent UA interactions, severely limiting its ability to optimize lifelong value. This confirms ASA is not optional, but essential for industrial-scale deployment.

To further validate generalizability, we deployed RLIV-UA on the Kuaishou short-video feed from November to December 2024, again using a 20% user cohort for treatment. Results show statistically significant improvements in long-term user engagement: +0.131% in average watch time, +0.083% in daily app usage duration, and +0.021% in weekly retention rate. Notably, in a system of Kuaishou's scale, even a 0.02% gain in retention is considered highly significant, reflecting millions of additional active users. These results confirm that RLIV-UA not only drives immediate revenue growth but also fosters sustainable, long-term platform health by strengthening user-author relationships.

5 RELATED WORK

5.1 RL-Based Recommendation Systems

[35] is the earliest work that tries to alternate multitask learning ranking model with RL model using DQN to learn the value of all items in the recommended list. Similarly, Chen et al. [6] employ a policy-gradient approach in RS and Zhao et al. [48] develop an actor-critic approach for recommending a page of items. However, they are not applied in a real-world recommendation environment with large amount of users and items. Then, more research [14, 49] aims to apply the RL model in reality as a substitute with simple network structure. In order to handle the huge number of candidate items, SlateQ [19] is proposed to decompose the value of item list into the sum of value of each item under some assumptions. Other literatures [9, 27] use contrastive learning to overcome the curse of dimensionality whose model structures are more complex.

5.2 Long-Term User Engagement in Recommendation Systems

In order to consider the long-term user engagement rather than user's immediate feedback, some research has increasingly focused on the sequential patterns of user behavior by employing temporal models, such as hidden Markov models and recurrent neural networks [5, 17, 32, 33, 39, 43]. Besides, some research [14, 34, 38] use RL to make a long-term planning. However, all the methods are too complex to be applied in practice. [51] propose a hierarchical LSTM based Q network to model the complex user behavior and design an S-network to simulate the environment avoiding the instability. [7] inspired by exploration research [3, 30] in RL use a series of exploration methods to improve user experience. [42] carefully design the reward function through data analysis to connect the long-term rewards with immediate feedback. While [44] propose a framework for learning preferences from user historical behavior sequences, specifically using preferences to automatically train a reward function in an end-to-end manner. Considering all the above methods' action is to select an item list which may be not practical when the number of item and user is large, [4] aim to optimize the weights of each predicted user feedback when ranking items under the long-term rewards with designed heuristic rewards to overcome the latency and sparsity of long-term rewards.

6 CONCLUSION

In this paper, we propose a novel lifelong interaction value model for user-author pairs, i.e. RLIV-UA, based on RL. Firstly, the interactions of UA pairs via RS is modeled as a sparse cross-request interaction markov decision process. To solve the long time interval and large scale of UA's interactons, an adjacent state approximation method is designed to build the RL training sample. Besides, to capture the progressive changes of lifelong UA relationship, a multi-task critic learning architecture is employed to utilize denser interaction signals to compensate for sparse labels. Moreover, an auxiliary supervised learning task is designed to improve the convergence of the RLIV-UA model in large-scale RS. Finally, in both offline environments and online A/B tests, the experiment results show that the proposed RLIV-UA model performs better under both user satisfaction metrics and author benefits metrics, resulting in higher platform profits, compared with other models.

As future work, we plan to explore the application of RLIV-UA's core components, particularly the SCRI-MDP formulation, ASA method, and MTCL architecture , in other domains such as e-commerce (user-merchant) and music streaming (user-artist), where optimizing sparse, long-term, bidirectional value is equally critical. This will require careful adaptation of reward functions and interaction definitions to fit each domain's unique dynamics.

REFERENCES

[1] M Mehdi Afsar, Trafford Crump, and Behrouz Far. 2022. Reinforcement learning based recommender systems: A survey. *Comput. Surveys* 55, 7 (2022), 1–38.

[2] Amos Azaria, Avinatan Hassidim, Sarit Kraus, Adi Eshkol, Ofer Weintraub, and Irit Netanely. 2013. Movie recommender system for profit maximization. In *Proceedings of the 7th ACM conference on Recommender systems*. 121–128.

[3] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. 2016. Unifying count-based exploration and intrinsic

motivation. *Advances in neural information processing systems* 29 (2016).

[4] Qingpeng Cai, Shuchang Liu, Xueliang Wang, Tianyou Zuo, Wentao Xie, Bin Yang, Dong Zheng, Peng Jiang, and Kun Gai. 2023. Reinforcing user retention in a billion scale short video recommender system. In *Companion Proceedings of the ACM Web Conference 2023*. 421–426.

[5] Pedro G Campos, Fernando Díez, and Iván Cantador. 2014. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction* 24 (2014), 67–119.

[6] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. 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.

[7] Minmin Chen, Yuyan Wang, Can Xu, Ya Le, Mohit Sharma, Lee Richardson, Su-Lin Wu, and Ed Chi. 2021. Values of user exploration in recommender systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 85–95.

[8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[9] Romain Deffayet, Thibaut Thonet, Jean-Michel Renders, and Maarten De Rijke. 2023. Generative slate recommendation with reinforcement learning. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 580–588.

[10] M Benjamin Dias, Dominique Locher, Ming Li, Wael El-Deredy, and Paulo JG Lisboa. 2008. The value of personalised recommender systems to e-business: a case study. In *Proceedings of the 2008 ACM conference on Recommender systems*. 291–294.

[11] Scott Fujimoto and Shixiang Shane Gu. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems* 34 (2021), 20132–20145.

[12] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*. PMLR, 1587–1596.

[13] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. 3953–3957. https://doi.org/10.1145/3511808.3557624

[14] Jason Gauci, Edoardo Conti, Yitao Liang, Kittipat Virochsiri, Yuchen He, Zachary Kaden, Vivek Narayanan, Xiaohui Ye, Zhengxing Chen, and Scott Fujimoto. 2018. Horizon: Facebook's open source applied reinforcement learning platform. *arXiv preprint arXiv:1811.00260* (2018).

[15] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. 2020. Hierarchical user profiling for e-commerce recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 223–231.

[16] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2019. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603* (2019).

[17] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.

[18] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the eighth international workshop on data mining for online advertising*. 1–9.

[19] Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Tushar Chandra, and Craig Boutilier. 2019. SlateQ: A tractable decomposition for reinforcement learning with recommendation sets. (2019).

[20] Dietmar Jannach and Gediminas Adomavicius. 2017. Price and profit awareness in recommender systems. *arXiv preprint arXiv:1707.08029* (2017).

[21] Dietmar Jannach and Christine Bauer. 2020. Escaping the McNamara fallacy: Towards more impactful recommender systems research. *Ai Magazine* 41, 4 (2020), 79–95.

[22] Dietmar Jannach and Michael Jugovac. 2019. Measuring the business value of recommender systems. *ACM Transactions on Management Information Systems (TMIS)* 10, 4 (2019), 1–23.

[23] Mathias Jesse and Dietmar Jannach. 2021. Digital nudging with recommender systems: Survey and future directions. *Computers in Human Behavior Reports* 3 (2021), 100052.

[24] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems* 33 (2020), 1179–1191.

[25] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[26] Jingxin Liu, Xiang Gao, Yisha Li, Xin Li, Haiyang Lu, and Ben Wang. 2024. Supervised Learning-enhanced Multi-Group Actor Critic for Live Stream Allocation in Feed. *arXiv preprint arXiv:2412.10381* (2024).

[27] Shuchang Liu, Qingpeng Cai, Bowen Sun, Yuhao Wang, Ji Jiang, Dong Zheng, Peng Jiang, Kun Gai, Xiangyu Zhao, and Yongfeng Zhang. 2023. Exploration and regularization of the latent action space in recommendation. In *Proceedings of the ACM Web Conference 2023*. 833–844.

[28] Ziru Liu, Jiejie Tian, Qingpeng Cai, Xiangyu Zhao, Jingtong Gao, Shuchang Liu, Dayou Chen, Tonghao He, Dong Zheng, Peng Jiang, et al. 2023. Multi-task recommendations with reinforcement learning. In *Proceedings of the ACM web conference 2023*. 1273–1282.

[29] Wei Lu, Shanshan Chen, Keqian Li, and Laks VS Lakshmanan. 2014. Show me the money: Dynamic recommendations for revenue maximization. *Proceedings of the VLDB Endowment* 7, 14 (2014), 1785–1796.

[30] Volodymyr Mnih. 2016. Asynchronous Methods for Deep Reinforcement Learning. *arXiv preprint arXiv:1602.01783* (2016).

[31] Zbigniew W Ras, Katarzyna A Tarnowska, Jieyan Kuang, Lynn Daniel, and Doug Fowler. 2017. User friendly NPS-based recommender system for driving business revenue. In *Rough Sets: International Joint Conference, IJCRS 2017, Olsztyn, Poland, July 3–7, 2017, Proceedings, Part I*. Springer, 34–48.

[32] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[33] Nachiketa Sahoo, Param Vir Singh, and Tridas Mukhopadhyay. 2012. A hidden Markov model for collaborative filtering. *MIS quarterly* (2012), 1329–1356.

[34] Guy Shani, David Heckerman, Ronen I Brafman, and Craig Boutilier. 2005. An MDP-based recommender system. *Journal of machine Learning research* 6, 9 (2005).

[35] Peter Sunehag, Richard Evans, Gabriel Dulac-Arnold, Yori Zwols, Daniel Visentin, and Ben Coppin. 2015. Deep reinforcement learning with attention for slate markov decision processes with high-dimensional states and actions. *arXiv preprint arXiv:1512.01124* (2015).

[36] Richard S Sutton. 2018. Reinforcement learning: An introduction. *A Bradford Book* (2018).

[37] Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112, 1-2 (1999), 181–211.

[38] Nima Taghipour, Ahmad Kardan, and Saeed Shiry Ghidary. 2007. Usage-based web recommendations: a reinforcement learning approach. In *Proceedings of the 2007 ACM conference on Recommender systems*. 113–120.

[39] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 17–22.

[40] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[41] Wenlin Wang. 2021. Learning to recommend from sparse data via generative user feedback. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4436–4444.

[42] Yuyan Wang, Mohit Sharma, Can Xu, Sriraj Badam, Qian Sun, Lee Richardson, Lisa Chung, Ed H Chi, and Minmin Chen. 2022. Surrogate for long-term user experience in recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 4100–4109.

[43] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 495–503.

[44] Wanqi Xue, Qingpeng Cai, Zhenghai Xue, Shuo Sun, Shuchang Liu, Dong Zheng, Peng Jiang, Kun Gai, and Bo An. 2023. PrefRec: recommender systems with human preferences for reinforcing long-term user engagement. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2874–2884.

[45] Eva Zangerle and Christine Bauer. 2022. Evaluating recommender systems: survey and framework. *Comput. Surveys* 55, 8 (2022), 1–38.

[46] Gengrui Zhang, Yao Wang, Xiaoshuang Chen, Hongyi Qian, Kaiqiao Zhan, and Ben Wang. 2024. UNEX-RL: reinforcing long-term rewards in multi-stage recommender systems with unidirectional execution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9305–9313.

[47] Qihua Zhang, Junning Liu, Yuzhuo Dai, Yiyan Qi, Yifan Yuan, Kunlun Zheng, Fan Huang, and Xianfeng Tan. 2022. Multi-task fusion via reinforcement learning for long-term user satisfaction in recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 4510–4520.

[48] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM conference on recommender systems*. 95–103.

[49] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 world wide web conference*. 167–176.

[50] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 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.

[51] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. 2019. Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2810–2818.

A PROOFS

A.1 Equivalence to augmented MDP

Theorem A.1 (Equivalence to augmented MDP). *Consider a Sparse Cross-Request Interaction MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \Delta \rangle$ with interaction gap variable $\Delta^t_{\text{ua}} \geq 1$. Define an augmented state $\bar{s}_t = (s^t_{\text{ua}}, \Delta^t_{\text{ua}})$ where Δ^t_{ua} is treated as part of the state vector, forming an augmented state space $\bar{\mathcal{S}} = \mathcal{S} \times \mathbb{N}$. Then $\mathcal{M}$ is equivalent to a standard MDP $\bar{\mathcal{M}} = \langle \bar{\mathcal{S}}, \mathcal{A}, \bar{\mathcal{P}}, \mathcal{R}, \gamma \rangle$, where the transition kernel is defined as:*

$$\bar{P}((\mathbf{s}', \Delta') \mid (\mathbf{s}, \Delta), a) = P(\mathbf{s}'_{\text{ua}}, \Delta'_{\text{ua}} \mid \mathbf{s}_{\text{ua}}, \Delta_{\text{ua}}, a_{\text{ua}}), \tag{16}$$

and the reward function $\mathcal{R}$ remains identical. Any optimal policy π^* in $\mathcal{M}$ corresponds to an optimal policy in $\bar{\mathcal{M}}$ under this mapping.

Proof. We aim to prove that an optimal policy in the SCRI-MDP $\mathcal{M}$ corresponds to an optimal policy in the augmented MDP $\bar{\mathcal{M}}$. We proceed via **proof by contradiction**.

Let $\pi^* : \mathcal{S} \times \mathbb{N} \rightarrow \mathcal{A}$ be an optimal deterministic policy for the SCRI-MDP $\mathcal{M}$. By definition, for all $(s, \Delta) \in \mathcal{S} \times \mathbb{N}$ and for any other policy π, the value function satisfies:

$$V^{\pi^*}(s, \Delta) \geq V^{\pi}(s, \Delta) \tag{17}$$

Now, consider the corresponding policy $\bar{\pi}^*$ in the augmented MDP $\bar{\mathcal{M}}$, defined such that $\bar{\pi}^*(a|\bar{s}) = \pi^*(a|s, \Delta)$ for $\bar{s} = (s, \Delta)$. Assume, for the sake of contradiction, that $\bar{\pi}^*$ is **not** optimal in $\bar{\mathcal{M}}$. This implies that there exists another policy $\bar{\pi}' : \bar{\mathcal{S}} \rightarrow \mathcal{A}$ and some augmented state $\bar{s}_0 = (s_0, \Delta_0)$ such that:

$$V^{\bar{\pi}'}(s_0, \Delta_0) > V^{\bar{\pi}^*}(s_0, \Delta_0) \tag{18}$$

Define a policy π' for the original SCRI-MDP $\mathcal{M}$ by $\pi'(a|s, \Delta) := \bar{\pi}'(a \mid (s, \Delta))$. Since both MDPs share the same reward function $\mathcal{R}$, discount factor γ, and the dynamics of $\mathcal{M}$ are fully encoded in $\bar{\mathcal{P}}$ through the inclusion of Δ in the state, the expected return from any initial state-action pair is identical under corresponding policies. Specifically, the Bellman equations for both frameworks satisfy:

For the SCRI-MDP:

$$Q^{\pi}(s, \Delta, a) = \mathcal{R}(s, \Delta, a) + \gamma \sum_{s', \Delta'} P(s', \Delta' \mid s, \Delta, a) \sum_{a'} \pi(a' \mid s', \Delta') Q^{\pi}(s', \Delta', \pi) \tag{19}$$

For the augmented MDP:

$$Q^{\bar{\pi}}((s, \Delta), a) = \mathcal{R}(s, \Delta, a) + \gamma \sum_{s', \Delta'} \bar{P}((s', \Delta') \mid (s, \Delta), a) \sum_{a'} \bar{\pi}(a' \mid s', \Delta') Q^{\bar{\pi}}((s', \Delta'), \bar{\pi})) \tag{20}$$

Since $\bar{P}((s', \Delta') \mid (s, \Delta), a) = P(s', \Delta' \mid s, \Delta, a)$ by construction, and the reward function is identical, it follows that:

$$V^{\pi'}(s, \Delta) = V^{\bar{\pi}'}(s, \Delta) \quad \text{and} \quad V^{\pi^*}(s, \Delta) = V^{\bar{\pi}^*}(s, \Delta) \tag{21}$$

for all $(s, \Delta) \in \mathcal{S} \times \mathbb{N}$.

Substituting these identities into our earlier inequality yields:

$$V^{\pi'}(s_0, \Delta_0) > V^{\pi^*}(s_0, \Delta_0) \tag{22}$$

This contradicts the assumption that π^* is optimal for $\mathcal{M}$. Therefore, the assumption that $\bar{\pi}^*$ is not optimal in $\bar{\mathcal{M}}$ must be false. Hence, $\bar{\pi}^*$ is indeed an optimal policy for $\bar{\mathcal{M}}$.

Conversely, let $\bar{\pi}^*$ be an optimal deterministic policy for the augmented MDP $\bar{\mathcal{M}}$. Then, we have:

$$V^{\bar{\pi}^*}(\bar{s}) \geq V^{\bar{\pi}}(\bar{s}) \tag{23}$$

for all $\bar{s} \in \bar{\mathcal{S}}$ and for any other policy $\bar{\pi}$.

Define a policy π^* for the SCRI-MDP $\mathcal{M}$ by $\pi^*(a|s, \Delta) := \bar{\pi}^*(a \mid (s, \Delta))$. Assume, for contradiction, that π^* is not optimal in $\mathcal{M}$. Then there exists another policy π' and some state (s_0, Δ_0) such that:

$$V^{\pi'}(s_0, \Delta_0) > V^{\pi^*}(s_0, \Delta_0) \tag{24}$$

Define $\bar{\pi}'$ for $\bar{\mathcal{M}}$ by $\bar{\pi}'(a \mid (s, \Delta)) := \pi'(a \mid s, \Delta)$. Using the same argument as before, we have:

$$V^{\bar{\pi}'}(s_0, \Delta_0) > V^{\bar{\pi}^*}(s_0, \Delta_0) \tag{25}$$

which contradicts the optimality of $\bar{\pi}^*$ in $\bar{\mathcal{M}}$. Therefore, π^* must be optimal in $\mathcal{M}$.

This establishes a one-to-one correspondence between optimal policies in $\mathcal{M}$ and $\bar{\mathcal{M}}$. Consequently, the SCRI-MDP $\mathcal{M}$ is equivalent to the standard MDP $\bar{\mathcal{M}}$ under the augmented state formulation, completing the proof. □

A.2 Error Bound for ASA

Theorem A.2 (Error Bound for Approximate State Transitions). *Let Q^* be the optimal Q-function of the SCRI-MDP $\mathcal{M}$, and let $\hat{Q}$ be the Q-function learned using the ASA with maximum prediction error $\epsilon = \max_{s,a,r} \left|\hat{s}^{t+1} - s^{t+1}\right|$. If the ASA error ϵ is bounded and the reward function is Lipschitz continuous with constant L_r, then the difference between $\hat{Q}$ and Q^* is bounded by:*

$$\left|\hat{Q}(s, a) - Q^*(s, a)\right| \leq \frac{L_r \epsilon}{(1 - \gamma)^2} \tag{26}$$

for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, where $\gamma \in [0, 1)$ is the discount factor.

Proof. Let's denote the Bellman operator for the true MDP as $\mathcal{T}$, and for the approximate MDP with ASA as $\hat{\mathcal{T}}$. The Bellman operators are defined as:

$$\begin{aligned} \mathcal{T} Q(s, a) &= r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\max_{a'} Q(s', a') \right] \\ \hat{\mathcal{T}} Q(s, a) &= r(s, a) + \gamma \mathbb{E}_{\hat{s}' \sim \hat{P}(\cdot|s,a)} \left[\max_{a'} Q(\hat{s}', a') \right] \end{aligned} \tag{27}$$

where P is the true transition probability and $\hat{P}$ is the approximate transition probability induced by ASA.

The fixed points of these operators are the optimal Q-function Q^* and the approximate Q-function $\hat{Q}$, respectively:

$$\begin{aligned} Q^* &= \mathcal{T} Q^* \\ \hat{Q} &= \hat{\mathcal{T}} \hat{Q} \end{aligned} \tag{28}$$

We want to bound $\left|\hat{Q}(s, a) - Q^*(s, a)\right|$. Consider:

$$\begin{aligned}\left|\hat{Q}(s,a) - Q^*(s,a)\right| &= \left|\hat{\mathcal{T}}\hat{Q}(s,a) - \mathcal{T}Q^*(s,a)\right| \\ &\leq \left|\hat{\mathcal{T}}\hat{Q}(s,a) - \hat{\mathcal{T}}Q^*(s,a)\right| + \left|\hat{\mathcal{T}}Q^*(s,a) - \mathcal{T}Q^*(s,a)\right|\end{aligned} \tag{29}$$

The first term can be bounded using the contraction property of the Bellman operator:

$$\left|\hat{\mathcal{T}}\hat{Q}(s,a) - \hat{\mathcal{T}}Q^*(s,a)\right| \leq \gamma \left|\hat{Q} - Q^*\right| \tag{30}$$

For the second term, we have:

$$\begin{aligned}&\left|\hat{\mathcal{T}}Q^*(s,a) - \mathcal{T}Q^*(s,a)\right| \\ &= \gamma \left|\mathbb{E}_{\hat{s}' \sim \hat{P}(\cdot|s,a)}\left[\max_{a'} Q^*(\hat{s}',a')\right] - \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[\max_{a'} Q^*(s',a')\right]\right|\end{aligned} \tag{31}$$

Since Q^* is Lipschitz continuous with constant $L_Q = \frac{L_r}{1-\gamma}$ (a standard result in RL), and the ASA error is bounded by ϵ, we have:

$$\left|\hat{\mathcal{T}}Q^*(s,a) - \mathcal{T}Q^*(s,a)\right| \leq \gamma L_Q \epsilon = \gamma \frac{L_r}{1-\gamma}\epsilon \tag{32}$$

Combining these results:

$$\left|\hat{Q}(s,a) - Q^*(s,a)\right| \leq \gamma \left|\hat{Q} - Q^*\right| + \gamma \frac{L_r}{1-\gamma}\epsilon \tag{33}$$

Taking the supremum over all (s,a):

$$\begin{aligned}\left|\hat{Q} - Q^*\right| &\leq \gamma \left|\hat{Q} - Q^*\right| + \gamma \frac{L_r}{1-\gamma}\epsilon \\ (1-\gamma)\left|\hat{Q} - Q^*\right| &\leq \gamma \frac{L_r}{1-\gamma}\epsilon \\ \left|\hat{Q} - Q^*\right| &\leq \frac{L_r \epsilon}{(1-\gamma)^2}\end{aligned} \tag{34}$$

This completes the proof. □

B EXPERIMENTAL SETTINGS AND ALGORITHM PSEUDOCODE

B.1 Evaluation Metrics

The detailed evaluation metrics of compared methods are shown as follows:

- **Session Length**: The number of requests in one session of a user with RS, which directly reflects the user satisfaction of the platform.
- **Watch Time**: The accumulated watching time of all items watched by a user in one session.
- **CTR**: The average click rate of all items recommended to a user in one session.
- **Diversity**: Quantifies the variety of content types in recommendations and is highly related to author benefits.
- **New Fans**: The total number of new followers accumulated by authors.
- **UA Count**: The count of user-author pairs with "deep" relationship which is defined as whether user has followed author, whether user has given author the most gifts, and other conditions.
- **Weekly Gifted Users**: The number of gifted user in a week and it indicates the revenue scale of users in the platform.
- **Total Revenue**: The important and ultimate metric to evaluate the platform revenue profits.
- **App Usage Time**: Average time users spend on the app.
- **Weekly Retention**: The stickness of user in a week.

B.2 Implementation Details

Notably, all the models adopt the same hyperparameters listed in Table 3 for fair comparison.

Table 3: Hyperparameters of the compared models.

Hyper-parameter	Value
Optimizer	Adam
γ Discount factor	0.9
τ Target network update rate	0.005
Learning rate of critic	1e-3
Learning rate of actor	1e-4
Batch size	1024
Train epochs	250
Hidden layer dimensions	[64, 64]
The dimension of embedding layer	32
Learning rate of embedding layer	1e-3
Training steps per epoch	1e4
Training Platform	PyTorch

B.3 Algorithm Pseudocode

As shown in Algorithm 1, the training process of the RLIV-UA is divided into two steps: The pre-training of the state predictor f_θ and the joint training of the LIV networks and the tate predictor.

C DATA ANALYSIS

Firstly, we observe that the lifelong UA interactions have strong connection with the ultimate platform revenue. As shown in Fig. 8a, the count of UA pairs with both follow and frequent gift ("deep") relationship is positively correlated with the total revenue value. On the other hand, as users continue to interact with the authors with "deep" relationship, its conversion rate will increase, as shown in Fig. 8b. Hence, it provides insights to improve the platform profits by modeling the LIV of UA pairs.

In particular, to more clearly evaluate how the RLIV-UA model optimizes the LIV of UA pairs a specific example is shown in Fig. 9. We collect the lifelong revenue value (Q value) output by the RLIV-UA model with progressive UA interactions from click, watch to interact, and gift. It shows that as the UA output lifelong revenue value of user-author pair is progressively increasing as their relationship becomes deeper.

D MORE DISCUSSION OF ASA

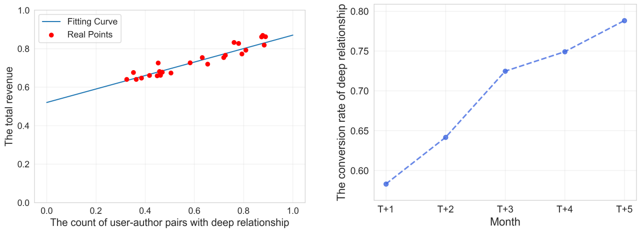
D.1 The rule-based approximation

The details of the rule-based approximation are as follows.

Specifically, the dynamic features $\mathbf{f}^t_{\text{ua}}$ in the current state $\mathbf{s}^t_{\text{ua}}$ is defined as the counts of several kinds of interactions between ua. Note that $\mathbf{f}^{t+1}_{\text{ua}}$ can be approximated based on user's current rewards

Algorithm 1 The RLIV-UA Algorithm

```
Input: Initial parameters of all the Q networks $\phi_i, i = 1, ..., n$, initial parameters of state predictor $\theta$, learning rate $\lambda$, batch size for updating $B$ and the update rate of target networks $\tau$
Initialize the target networks with $\phi'_i \leftarrow \phi_i$
Initialize the replay memory buffer of Q networks $D \leftarrow \emptyset$
Repeat:
Step1: The pre-training of state predictor
for epoch = 1 to train_epochs do
    for each interact_step $t$ of each user-author pair ua do
        Observe the current state $\mathbf{s}^t_{\text{ua}}$.
        Retrieve the last interaction from user historical behaviors sequences $\mathbf{s}^{t-1}_{\text{ua}}, a^{t-1}_{\text{ua}}, r^{t-1}_{\text{ua}}$.
        Conduct gradient update for $f_\theta$ :$\theta \leftarrow \theta - \lambda \nabla_\theta \mathcal{L}_\theta(\theta)$
    end for
end for
Step2: The join-training of LIV networks
for epoch = 1 to train_epochs do
    for each interact_step $t$ of each user-author pair ua do
        Observe the current state $\mathbf{s}^t_{\text{ua}}$, action $a^t_{\text{ua}}$ and rewards $r^t_{\text{ua},i}, i = 1, ..., n$.
        Input them into the state predictor $f_\theta$ to predict the next state $\hat{\mathbf{s}}^{t+1}_{\text{ua}}$.
        $D \leftarrow D \cup (\mathbf{s}^t_{\text{ua}}, a^t_{\text{ua}}, r^t_{\text{ua}}, \hat{\mathbf{s}}^{t+1}_{\text{ua}})$
        Retrieve the last interaction from user historical behaviors sequences $\mathbf{s}^{t-1}_{\text{ua}}, a^{t-1}_{\text{ua}}, r^{t-1}_{\text{ua}}$.
        Conduct gradient update for $f_\theta$ :$\theta \leftarrow \theta - \lambda \nabla_\theta \mathcal{L}_\theta(\theta)$
        if the number of samples in $D$ reaches $B$ then
            sample a batch from $D$ and calculate total loss $\mathcal{L}$ shown in 15
            $\phi_i \leftarrow \phi_i - \lambda \nabla_{\phi_i} \mathcal{L}(\phi_i)$
            $\phi'_i \leftarrow (1 - \tau)\phi_i + \tau\phi'_i$
        end if
    end for
end for
Return The final Q networks parameters
```



(a) The relationship between the count of UA pairs with "deep" relationship and the total revenue.

(b) The conversion rate of UA pairs with "deep" relationship.

Figure 8: The revenue relevance and conversion rate of the lifelong UA relationship in Kwai app. Note that all data is collected in the second half of 2024 from Kwai app and scaled between 0 and 1.

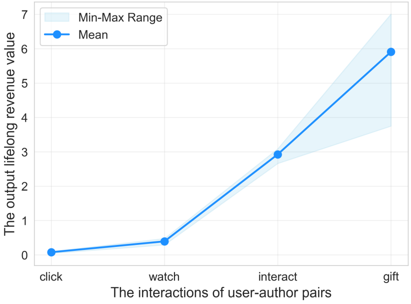


Figure 9: The output lifelong revenue values of some user-author pairs as their relationship becomes deeper.

r^t_{ua} and current dynamic UA features $\mathbf{f}^t_{\text{ua}}$:

$$\mathbf{f}^{t+1}_{ua} = \begin{cases} \mathbf{f}^t_{ua} + 1, & \text{if } r^t_{ua} > 0 \\ \mathbf{f}^t_{ua}, & \text{otherwise} \end{cases} \tag{35}$$

Since the SCRI-MDP only focuses on the interactions and state transitions between UA pair like ua, it ignores interactions between user u and items of other authors. Under the above assumption of the SCRI-MDP, the next state $\hat{\mathbf{s}}^{t+1}_{ua}$ can be derived by the dynamic features $\mathbf{f}^{t+1}_{ua}$.

D.2 Offline evaluation of ASA

Table 4: The evaluation loss of the adjacent state approximation compared with rule-based approximation.

Methods	MSE loss	Cosine Similarity loss
rule-based approximation	0.718	0.743
proposed ASA method	**0.502**	**0.630**

To demonstrate the effectiveness of the ASA method, we conduct a comparative evaluation against a rule-based baseline on an offline test set. The full dataset is randomly split into training and test sets in an 8:2 ratio. As shown in Table 4, the ASA method achieves significantly lower prediction error across all key state features, confirming its superior ability to model the complex, long-term dynamics of user-author interactions compared to the heuristic rule-based approach.