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 in short-video 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, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The recommendation system (RS) aims to explore users' interested content, while helping content authors reach potential target users to accumulate fans and obtain profits [1, 23, 44]. By promoting repeated interactions from both users and authors, RS helps build a more active ecosystem, leading to increased platform traffic and commercial returns [2, 10, 20–22, 29, 31].

In the current field of RS, some research focuses on improving the accuracy of user immediate feedback prediction [8, 15, 18, 25, 49], i.e. click-through rate (CTR), at each request by deep neural network (DNN) model, as shown in Fig. 1. Other studies concentrate on optimizing long-term engagement from **the user's perspective** [4, 7, 40, 41, 43, 46, 48, 50]. Specifically, they utilize reinforcement learning (RL) to dynamically optimize session-level long-term cumulative rewards [11, 36, 39]. However, the aforementioned research ignores the recommendation impact on authors and **the Lifelong Interaction Value (LIV) of user-author (UA) pairs**. Therefore, it is impossible to characterize the progressive changes of the lifelong UA relationship, which is essential for improving

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

ACM ISBN 978-1-4503-XXXX-X/2018/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

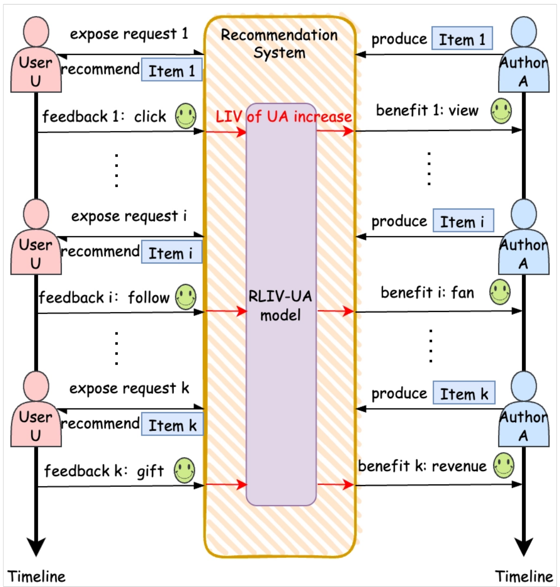
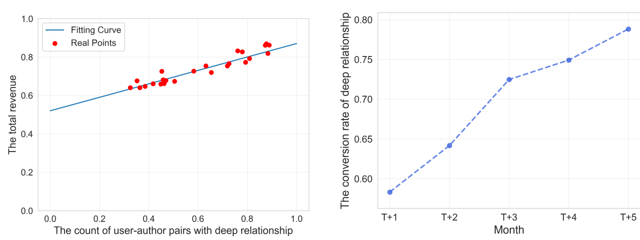


Figure 1: The main process of the proposed RLIV-UA model optimizing the LIV of UA pair based on their interactions.

the stickiness (retention) of UA pairs and the prosperity of social community in short-video platforms.



(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 2: 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.

Furthermore, we observe that the lifelong UA interactions have strong connection with the ultimate platform revenue. As shown in Fig. 2a, 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. 2b. Hence, it provides insights to improve the platform profits by modeling the LIV of UA pairs.

To the best of our knowledge, this paper is the first to **R**einforce **L**ifelong **I**nteraction **V**alue of **U**ser-**A**uthor pairs (RLIV-UA) using RL, based on each UA's interaction and its corresponding cumulative reward. As shown in Fig. 1, the RLIV-UA model dynamically optimizes the LIV of an UA pair to improve the stickiness of both user U and author A progressively.

However, directly applying RL to model the LIV of UA pairs presents several challenges: First, compared with existing request-based RL applications in RS, the UA interactions may happen in requests with long time span. In practice, industrial RSs involve massive numbers of users and authors, resulting in an enormous and sparse UA state space, which limits the storage of all interaction traces of UA pairs over a long time period, such as three month.

To solve the above problems, 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. Moreover, we propose a Multi-Task Critic Learning (MTCL) [28] architecture to model gradually stronger UA relationships, such as click → follow → gift. Finally, due to the sample sparsity of each UA pair and high variance of labels, an auxiliary supervised learning task is designed to improve the stability and convergence of the RLIV-UA model.

Overall, the main contributions of this paper are as follows:

- A novel LIV model, i.e. RLIV-UA, is proposed to reinforce the lifelong interaction value of UA pairs for large-scale RS.
- Based on SCRI-MDP, a novel ASA method is introduced to construct RL training samples.
- An MTCL architecture is proposed to model not only trivial interactions like click and watch time, but also "deep" interactions like follow and gift. To improve the convergence of the RLIV-UA model, an auxiliary supervised learning task is designed.
- Results in offline and online experiments show that the RLIV-UA model can improve both user engagement and author benefits, thus improving platform profits.
- The RLIV-UA has been successfully deployed on two short-video applications, i.e. Kuaishou and Kwai, which include over 400 million daily active users.

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 (SCRI-MDP) to model the LIV of UA pairs. Formally, it is represented by a tuple of five elements $< \mathcal{S}, \mathcal{A}, P, \mathcal{R}, \gamma >$:

- **State space** $\mathcal{S}$: The state $s_{ua} \in \mathcal{S}$ includes user u's static features (e.g., user ID, location, gender), item and author a's static features (e.g., item ID, author ID) and dynamic features of the ua pair (e.g. cumulative gift count, cumulative watch time).
- **Action space** $\mathcal{A}$: The action space $\mathcal{A}$ is defined as the candidate items in the ranking stage, and typically contains a

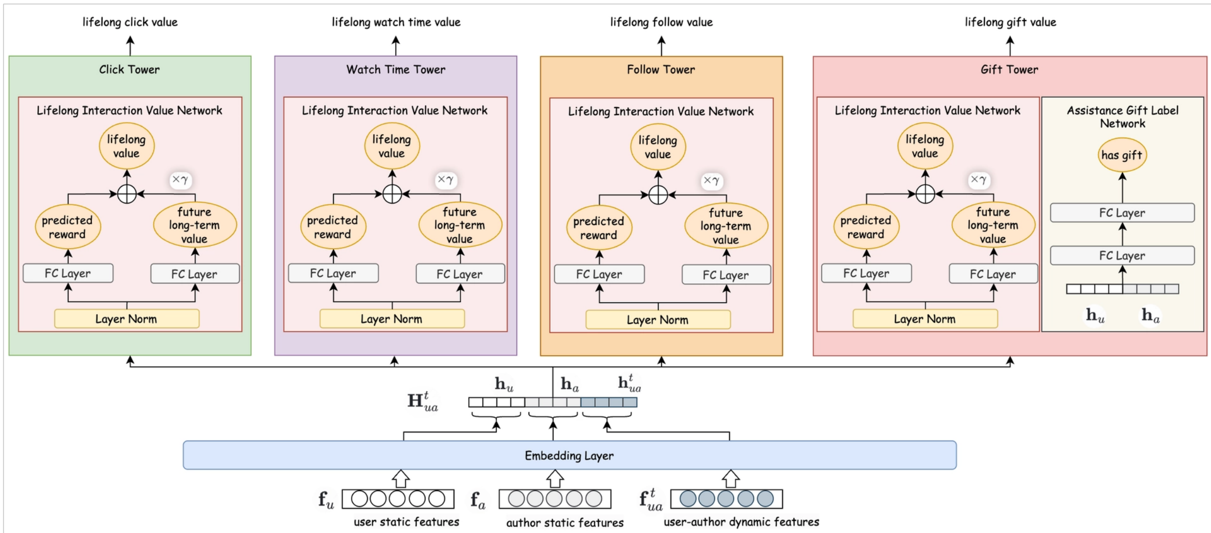


Figure 3: The overall framework of the proposed RLIV-UA model.

few hundred items. The corresponding action $a \in \mathcal{A}$ is a recommended item list.

- **State transition probability distribution** P: It is denoted as $P(\mathbf{s}'_{ua}|\mathbf{s}_{ua}, a)$ determined by the environment, where $\mathbf{s}'_{ua}$ represents the next state of the same ua pair.
- **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 $r_{ua,c}, r_{ua,w}, r_{ua,f}, r_{ua,g}$ be the reward function of click, watch time, follow and gift labels respectively. $C = \{c, w, f, g\}$ denotes the target label set and $n = 4$ denotes the cardinality of C.
- **Discount factor** γ: $\gamma \in [0, 1]$ is used to trade off instant rewards and future rewards.

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. The final objective function is as follows:

$$\max_{\phi_1,\cdots,\phi_n} \sum_{l \in C} \mathbb{E}\left[\sum_{t=0}^{\infty} r_{ua,l}^{t}\right] \tag{1}$$

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 I_i which is the t-th interaction of ua, then the LIV of ua pair is defined as:

$$Q(\mathbf{s}_{ua}^{t}, I_i) = r_{ua}^{t} + \gamma \max_{j \in \mathcal{I}} Q(\mathbf{s}_{ua}^{t+1}, I_j) \tag{2}$$

where $\mathcal{I}$ is the item set containing all items produced by author a.

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. Note that the overall framework of the RLIV-UA model is shown in Fig. 3.

3.1 Adjacent State Approximation

Since adjacent UA interactions happen in requests with a long time span, and the UA state space is extremely large with sparse UA interactions, the LIV of UA pairs is modeled as the SCRI-MDP. On one hand, the next state s' in the RL training sample (s, a, r, s') is not available until the next interaction of the same UA pair happens. On the other hand, it is impossible to store the interaction traces for all UA pairs in the industrial RS. Therefore, we design that the state space only contains user or author static features and the dynamic features between them:

$$\mathbf{s}_{ua}^{t} = \{\mathbf{f}_u, \mathbf{f}_a, \mathbf{f}_{ua}^{t}\} \tag{3}$$

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

$$\mathbf{f}_{ua}^{t+1} = \begin{cases} \mathbf{f}_{ua}^{t} + 1, & \text{if } r_{ua}^{t} > 0 \\ \mathbf{f}_{ua}^{t}, & \text{otherwise} \end{cases} \tag{4}$$

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 $\hat{\mathbf{s}}_{ua}^{t+1}$ can be derived by the dynamic features $\mathbf{f}_{ua}^{t+1}$.

3.2 Multi-Task LIV Networks

In order to model the progressive changes of lifelong UA relationship, we elaborately design 4 LIV networks based on the MTCL architecture, including Click, Watch time, Follow and Gift LIV networks. In particular, denser interaction signals (e.g. click and watch time) are used to compensate for sparse labels (e.g. follow and gift). For simplicity, we take one task tower as an example to explain the network structure.

As shown in Fig. 3, the current ua state $\{\mathbf{f}_u, \mathbf{f}_a, \mathbf{f}^t_{ua}\}$ is fed into a shared Embedding Layer to obtain corresponding hidden embeddings $\mathbf{h}_u$, $\mathbf{h}_a$ and $\mathbf{h}^t_{ua}$. Taking the vector $\mathbf{H}^t_{ua} = \text{concat}(\mathbf{h}_u, \mathbf{h}_a, \mathbf{h}^t_{ua})$ as the network input, the $Q_\phi(\mathbf{H}^t_{ua})$ is denoted as the final LIV of ua at the t-th ineraction. To mitigate the deviation problem of value overestimation [39], double value networks and two corresponding target networks are used to output the minimum value:

$$Q_\phi(\mathbf{H}^t_{ua}) = min(Q_{\phi^1}(\mathbf{H}^t_{ua}), Q_{\phi^2}(\mathbf{H}^t_{ua})) \tag{5}$$

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

$$\mathcal{L}(\phi) = \mathbb{E}_{(\mathbf{s}^t_{ua}, r^t_{ua}, \mathbf{s}^{t+1}_{ua})\in D}[(Q_\phi(\mathbf{H}^t_{ua}) - y)^2] \tag{6}$$

$$y = r^t_{ua} + \gamma \max_{j\in\mathcal{I}} Q'_{\phi'}(\mathbf{H}^{t+1}_{ua}) \tag{7}$$

where D indicates the real-time collected sample buffer, y indicates the target output value of the LIV network, and $Q'_{\phi'}$ represents the output value of target network with same structure as 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 sparse gift label of UA pairs in the Gift tower, an assistance neural network with two full-connected layers is designed to predict whether user u will gift author u at this interaction. As shown in Fig. 3, the user and author static hidden embeddings are input to the assistance network. And the binary cross entropy loss of the assistance gift binary classification goal is added to total loss.

3.3 Auxiliary Supervised Learning Network

Previous work [26] finds that the target value y is often dominated by the inaccurate output of the target network $Q'_{\phi'}(\mathbf{H}^{t+1}_{ua})$ 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:

$$Q_\phi(\mathbf{H}^t_{ua}) := \hat{r}_\eta(\mathbf{H}^t_{ua}) + \gamma \times \hat{V}_\phi(\mathbf{H}^t_{ua}) \tag{8}$$

where $\hat{r}_\eta(\mathbf{H}^t_{ua})$ represents the current predicted reward between ua at the t-th interaction with η as its trainable parameters, and $\hat{V}_\phi(\mathbf{H}^t_{ua})$ is the future cumulative reward value excluding the current reward.

As real r^t_{ua} is available based on the t-th interaction between ua, the predicted reward $\hat{r}_\eta(\mathbf{H}^t_{ua})$ can be learned by supervised loss. Incorporating with the aforementioned clipped double Q-learning [12] shown in Eq. 5, the auxiliary supervised learning network improves the convergence of the RLIV-UA model.

Hence, the general loss of a LIV network is defined as:

$$\mathcal{L}_Q = huber\_loss(\hat{r}_\eta(\mathbf{H}^t_{ua}), r^t_{ua}) + \sum_{2}^{k=1}(huber\_loss(Q_{\phi_k}(\mathbf{H}^t_{ua}), y) \tag{9}$$

where the first term denotes the loss for the auxiliary supervised learning network, and the second term denotes the original critic learning loss.

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

$$\mathcal{L} = \sum_{l\in C} \mathcal{L}^l_Q + \mathcal{L}^g_A \tag{10}$$

where $\mathcal{L}^g_A$ is the assistance gift loss and $\mathcal{L}^l_Q$ is the loss function, defined in Eq. 9, for each label in the target label set C, respectively. In practice, LIV scores 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.

3.4 Online Deployment

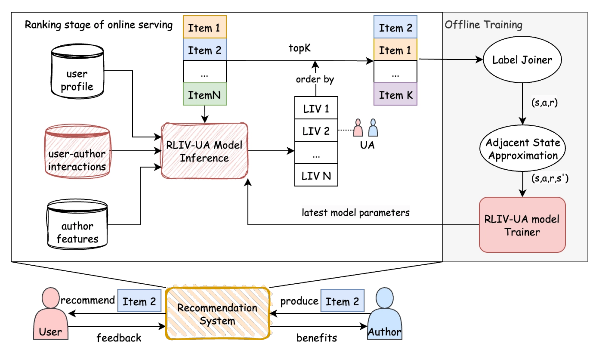


Figure 4: System architecture of the RLIV-UA model in real industrial recommendation scenario.

In real industrial RS, the online system architecture of RLIV-UA model is illustrated in Fig. 4, 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 s, action a, and the corresponding immediate feedback r at timestamp t. Then we leverage the ASA method to approximate the next state s' and obtain RL training sample (s, a, r, s'). In the online serving process, the RLIV-UA model is deployed 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 corresponding LIVs for N candidate items. The candidate items are then ranked based on their LIV scores. Finally, the top K candidate items are selected as the final item list and are exposed to the user by order. Subsequently, the user interacts with each exposure item to return the feedback signals to RS.

4 EXPERIMENTS

The proposed RLIV-UA model is evaluated in an offline simulated recommendation scenario to verify its performance compared with state-of-the-art models and the effectiveness of its each part. 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. The KuaiRand [13] is used as the offline experimental dataset to train an offline user simulator. It is a public recommendation dataset containing 27,285 users and 32,038,725 items obtained from Kuaishou app. Therefore, there are hundreds of millions of UA interactions. The trained user simulator is used as the offline environment to mimic the users' interaction with RS. In details, once simulator receives the recommended item, it returns immediate feedback like click, view and comments, etc. Then it determines whether to send the next request based on a quitting mechanism similar to that in work [45].

We evaluate the performance of compared methods in three aspects, including user satisfaction, author benefits and platform profits:

- **User Satisfaction**
  - **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.
- **Author Benefits**
  - **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.
- **Platform Profits**
  - **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.

4.1.2 Compared Methods. There are five models used as baseline in offline experiment and four variations of our model implemented in ablation experiment:

- **RankingModel**: The classic ranking model in RS. Specifically, the DeepFM [16] is applied in offline experiments.
- **CQL**[24]: An offline RL model that introduces a conservative constraint in the Q function update to limit the overly optimistic predictions of actions outside the data distribution.
- **DQN**[39]: A widely-used RL model that applies DNN as function approximator to estimate the Q-value of each action.
- **TD3**[11]: A classic RL model which uses twin critics to reduce the bias, delays the update of policy and smooths the synchrony of target networks.
- **RLUR**[4]: A RL model that aims 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.
- **RLIV-UA(w/o MT)**: The proposed RLIV-UA model without multi-task LIV netowrks.
- **RLIV-UA(w/o MT & SL)**: The proposed RLIV-UA model without multi-task LIV netowrks and auxiliary supervised learning task.
- **RLIV-UA(w/o AL & SL)**: The proposed RLIV-UA model without assistance gift label network and auxiliary supervised learning task.
- **RLIV-UA(w/o SL)**: The proposed RLIV-UA model without auxiliary supervised learning task.

4.1.3 Implementation Details. Notably, all the models adopt the same hyperparameters listed in Table 1 for fair comparison.

Table 1: 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

The assistance gift label network is not applied in offline experiments, because there is no gift signals in KuaiRand dataset.

Besides, 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 2. 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

Table 2: Overall performance of different 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	4.0610	49.5454	0.2247	0.7477
TD3	4.7820	52.7157	0.2228	0.8500
RLUR	6.6810	151.0550	0.4519	0.7206
RLIV-UA	**12.8860**	**377.4867**	0.5015	**0.8827**

Table 3: Overall performance of variations of the proposed RLIV-UA model in offline recommendation scenario.

Models	Session Length	Watch Time	CTR	Diversity
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**

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 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.

4.3 Ablation Study

The overall performance of RLIV-UA variations in offline recommendation scenario are shown in Table 3. 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. 5, 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.

4.4 Online A/B Experiments

Firstly, the proposed RLIV-UA model and its variants are deployed on Kwai live-stream feed with over 100 million users and 1 million authors to improve platform revenue from July to October 2024. Through online A/B experiments, the RLIV-UA(w/o AL & SL),

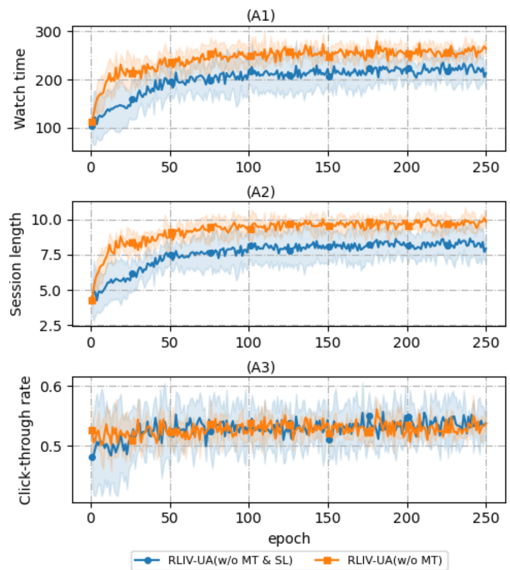


Figure 5: The learning process of RLIV-UA(w/o MT & SL) and RLIV-UA(w/o MT) over 10 rounds of training where the shaded areas correspond to the standard deviations.

RLIV-UA(w/o SL) and RLIV-UA models are successively evaluated under the platform profits metrics compared to RankingModel as baseline. As shown in Table 4, RLIV-UA variations successively perform better in both revenue metrics and new fans metric, which illustrates the practical effectiveness of all parts of RLIV-UA model.

Moreover, the output lifelong gift values corresponding to different grades of gift amount among UA pairs are shown in Fig. 6, illustrating the high positive correlation between the output lifelong gift value and actual revenue.

The proposed RLIV-UA model is also deployed on Kuaishou short-video feed to improve long-term user engagement from November to December 2024. The results are listed in Table 5, which shows that the RLIV-UA model can improve the app usage time

Table 4: 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)	+2.002%	+2.245%	+11.903%	+2.743%
RLIV-UA(w/o SL)	+3.579%	+3.417%	+23.8065%	+6.104%
RLIV-UA	**+5.849%**	**+5.370%**	**+38.6445%**	**+8.282%**

Table 5: Results of the watch time experiment on Kuaishou short-video feed.

Models	Watch Time	App Usage Time	Weekly Retention
RankingModel	-	-	-
RLIV-UA	**+0.131%**	**+0.083%**	**+0.021%**

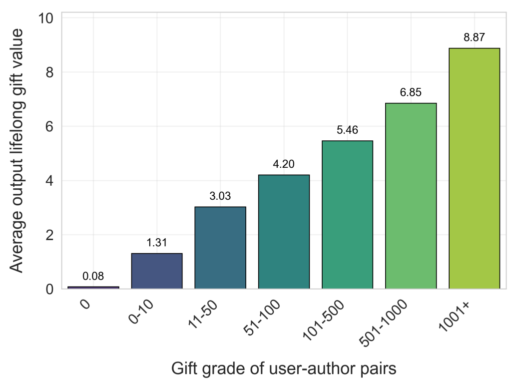


Figure 6: The output lifelong gift values of different grades of gift amount among UA pairs.

by 0.083% and the weekly retention by 0.021%. It should be emphasized that a 0.02% improvement holds statistical significance in the system. It is proved that the RLIV-UA model can not only improve platform revenue but also improve users' long-term retention.

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. [47] 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, 48] 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. Recently, some literatures [9, 27] use contrastive learning to overcome the curse of dimensionality whose model structures are relatively 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, 38, 42]. Besides, some research [14, 34, 37] use RL to make a long-term planning. However, all the methods are too complex to be applied in practice. Zou et al. [50] 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. Chen et al. [7] inspired by exploration research [3, 30] in RL use a series of exploration methods to improve user experience. Wang et al. [41] carefully design the reward function through data analysis to connect the long-term rewards with immediate feedback. While Xue et al. [43] 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, Cai et al. [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.

7 GENAI USAGE DISCLOSURE

We guarantee that no GenAI tools were used in any stage of the research, nor in the writing.

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] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[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] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

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

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.