ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 3 - 2026

Application of Reinforcement Learning in Optimizing New Media Content Recommendation Systems

Ruofei Gu[1,*]

[1] School of Digital Media and Design Arts, Beijing University of Posts and Telecommunications, Beijing, 102206, China

SUMMARY: *To address the conflict between immediate click-through rate and long-term retention in new media recommendation, this paper proposes DRL-MOREC, a deep reinforcement learning framework. A hybrid state encoder fuses users' short-term behavior sequences with long-term interest graphs to capture dual temporal scales of interest. A two-stage reward function allocates session-level 7-day retention prediction signals to each step via a discount factor, alleviating delayed reward sparsity. Conservative Q-learning and inverse propensity score weighting are introduced to mitigate distribution shift and popularity bias, respectively. Offline experiments on a short-video platform dataset show that the proposed method achieves a 7-day retention rate 2.0, 2.7, and 1.3 percentage points higher than DeepFM, DDPG-TD3, and SAC-Rec, respectively, while improving catalog coverage (ECC) by 0.09. Online A/B testing demonstrates an 11.3% lift in daily active user retention over DeepFM. Ablation studies reveal that the delayed reward contributes 4.0 percentage points to the retention improvement. These results validate the effectiveness of reinforcement learning in optimizing long-term user value under dynamic recommendation scenarios.*

KEYWORDS: *Deep Reinforcement Learning; Recommender System; New Media Content; Multi-Objective Optimization; User Retention*

1 Introduction

Recommendation systems in new media content platforms (such as short videos, picture and text information flows, live broadcasts) face a series of unique challenges. Different from the relatively stable commodity preferences in e-commerce, users' interests on new media platforms show significant short-term volatility: a popular video can change users' browsing trajectories within hours, and the next day users' interests may completely shift to another topic. This high dynamism makes it difficult for collaborative filtering methods based on static user portraits to adapt to the rapid drift of current intentions [1]. In contrast, deep learning ranking models (such as DeepFM, Wide&Deep) have made remarkable progress in CTR prediction tasks by introducing feature crossing and deep networks [2, 3], but the optimization objectives of these models are limited to the single-step click probability and cannot model the long-term evolution of user satisfaction.

Reinforcement learning has gradually been introduced into the field of recommendation systems due to its ability of sequential decision-making and delayed reward modeling [4]. Regarding the recommendation process as a Markov decision process, the policy can select actions according to the current state at each step and optimize the long-term goal through

*guruofei@bupt.edu.cn
https://doi.org/10.65102/is20261253
Published: 10 June, 2026

cumulative discounted rewards. Existing studies have shown that value-based reinforcement learning methods have improved in terms of user stay duration compared to supervised learning methods [5]. However, directly applying standard reinforcement learning algorithms to recommendation scenarios still faces several obstacles. Distribution shift problem during offline training: Log data is generated by the behavior policy, and there is a deviation between the current policy and it, resulting in overestimation of Q values [6]. Sparsity and delay of reward signals: Whether users retain often takes several days to observe, and the immediate click signal may mislead the policy to pursue short-term metrics at the expense of long-term value [7]. Popularity bias: Popular content gets more exposure in the logs, and its valuation is artificially inflated, leading the policy to tend to repeatedly recommend homogeneous content [8].

Regarding the above problems. In terms of distribution shift, conservative Q-learning suppresses overestimation by imposing penalties on unvisited actions [9], and this idea is introduced into the offline training of recommendation. In terms of delayed rewards, the practice of using the user retention prediction model as an additional reward signal has been verified to be effective [10], but the retention signal is usually given only at the end of a session, resulting in sparse single-step rewards. In terms of diversity optimization, the strategy of eliminating exposure bias through inverse propensity score weighting has been proven to improve the recommendation category coverage [11]. However, existing work often only focuses on one of the above problems and has not yet formed a unified framework that can simultaneously address dynamic state representation, delayed rewards, and popularity bias.

This paper proposes the DRL-MOREC framework, which attempts to jointly solve the above three problems under a unified Markov decision process framework. The core of this method lies in capturing the dual time scales of user interests through a hybrid state encoder (fusing short-term Transformer sequence encoding and long-term GraphSAGE graph encoding); designing a two-stage reward function to distribute the session-level retention prediction signal to each step through a discount factor to alleviate the sparsity of delayed rewards; introducing conservative Q learning and inverse propensity score weighting to respectively suppress the overestimation of out-of-distribution Q values and popularity bias. Offline and online experiments on the short video platform dataset show that this framework outperforms the current mainstream supervised learning and reinforcement learning baselines in terms of long-term retention and diversity metrics.

2 Related Work

2.1 Reinforcement Learning Recommendation Systems

The core motivation for introducing reinforcement learning into the recommendation field is its sequential decision-making ability, the user's satisfaction not only depends on the current recommended content but is also affected by the historical recommendation sequence. Early work applied deep Q-networks to news recommendation and proved that value-based reinforcement learning methods are superior to supervised learning baselines in improving user dwell time by modeling user browsing behavior as state transitions [12]. Subsequent research extended the action space to a page-level recommendation list and used policy gradient methods to optimize the overall quality of the list, alleviating the sub-optimal problem caused by greedy pointwise ranking [13]. However, these works all use immediate feedback (clicks, dwell time) as the only reward signal and do not consider the latency of user retention. Another study improved the estimation accuracy of policy gradients during offline training by introducing top-k off-policy correction in the REINFORCE algorithm, but its reward design is still limited to

single-step click-through rates [14]. Compared with the above work, this paper attempts to incorporate user retention prediction as a delayed reward into the optimization objective and alleviate the sparsity problem of rewards through a discount allocation mechanism.

2.2 Multi-objective Recommendation Optimization

Multi-objective optimization in recommendation systems involves the trade-off of multiple metrics such as click-through rate, dwell time, and diversity. A common approach is to transform multiple objectives into a single objective through weighted summation, for example, in a multi-task learning framework, different tasks' information is passed through sharing underlying parameters [15]. However, multi-task learning is essentially a static weighting of the losses of each objective and cannot handle the changes in objective priorities in a dynamic environment. Another approach is the method based on Pareto optimization, which provides multiple sets of candidate policies for decision-makers to choose by finding solutions on the Pareto front [16], but such methods have high computational costs and are difficult to adjust in real-time in an online environment. In the reinforcement learning framework, accumulating multiple reward signals into a scalar return is a natural way, but how to design the weights of each component to balance short-term and long-term objectives remains an open question [17]. In this paper, through the design of a two-stage reward function, immediate completion reward and delayed retention reward, the synergistic optimization of short-term engagement and long-term retention is achieved under a unified framework, and the complementary relationship between the two is verified through ablation experiments.

2.3 Bias Correction in Recommendations

Position bias and popularity bias are prevalent in new media recommendations. Position bias means that users tend to click on items at the top of the list, resulting in an overestimated click-through rate. The inverse propensity scoring (IPS) method corrects this bias by re-weighting the log feedback according to the probability of the item being observed [18]. Popularity bias refers to the fact that popular content obtains a higher click-through rate estimate due to more exposure, while unpopular content is systematically underestimated. Research from a causal perspective proposes estimating the true attractiveness of an item through counterfactual reasoning to separate the popularity factor [19]. In reinforcement learning recommendations, bias correction faces greater challenges: the exploration behavior of the policy changes the data distribution, and the bias in the log data is amplified through Q-value estimation. Conservative Q-learning suppresses overestimation by penalizing the Q-values of out-of-distribution actions in the loss function, but it does not explicitly handle popularity bias [20]. In this paper, based on conservative Q learning, IPS weighting is introduced, and the exposure propensity is embedded as a sample weight into Q-value updates, thereby correcting the systematic interference of popularity bias on value estimation while suppressing distribution shift.

3 Our Method (Methodology)

To address the inherent conflict between immediate click-through rate and long-term retention in new media content recommendations, and the contradiction that users' interests exhibit both short-term fluctuations and long-term structures on the time scale, as shown in Figure 1 this paper starts from the definition of the Markov decision process and constructs five modules in sequence: hybrid state encoding, flexible policy learning, two-stage reward function, offline

conservative regularization and causal bias correction, and online fine-tuning and safe exploration.

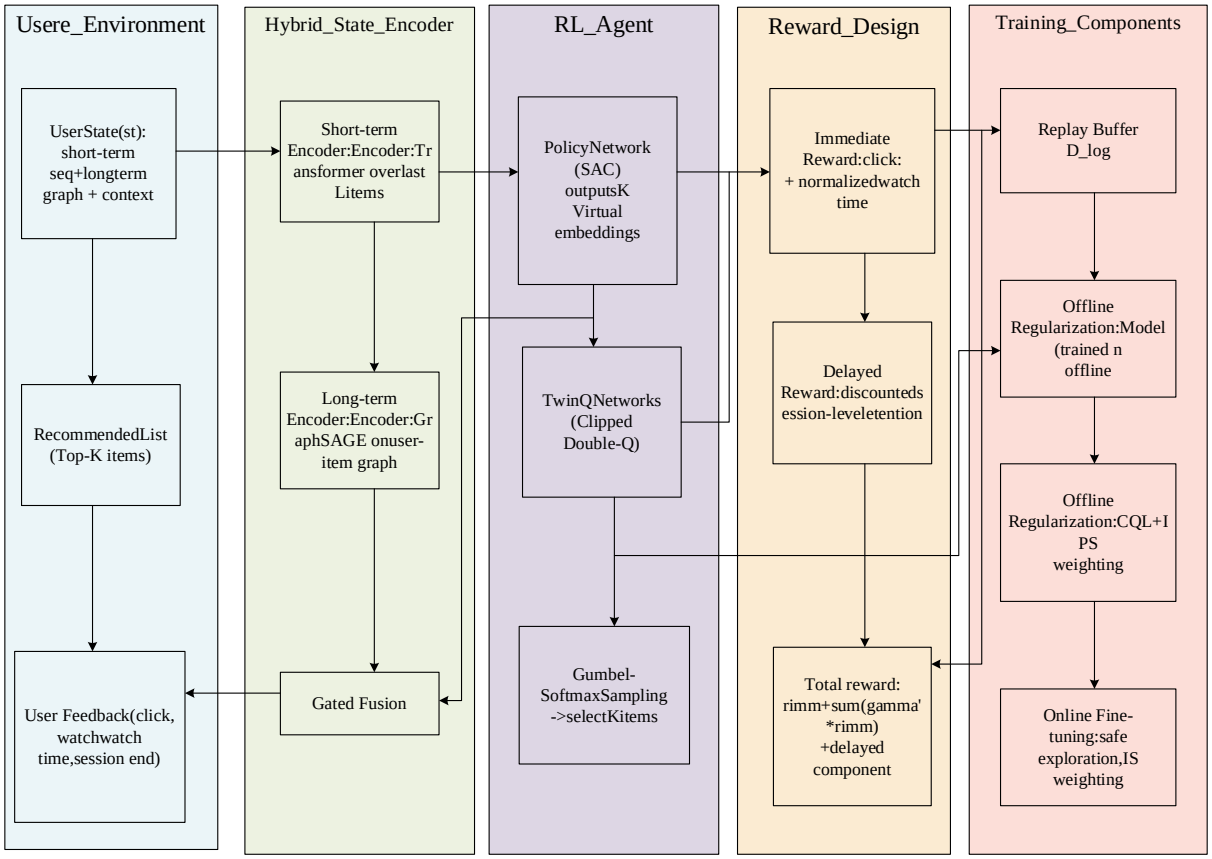

Figure 1: the overall framework of DRL-MOREC

3.1 Problem Formalization and MDP Modeling

Model the interaction process between users and the recommendation system as a discrete-time Markov decision process, denoted as the five-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. This modeling assumes that the user's next behavior depends only on the current state, including the recent interaction sequence, user intrinsic attributes, and context features, and is independent of more distant historical conditions. Although this Markovian assumption is too simplistic in a strict sense, it has been widely verified as an effective engineering approximation in the field of recommendation systems.

The design of the state space S determines the basis of the policy representation ability. At time t, the state s_t is composed of three parts: the encoded representation of the item sequence of the user's most recent L interactions, the user's long-term static portrait vector p_u (including the embedded aggregation of categorical features such as age, gender, and device), and the current context features c_t (such as the sine/cosine encoding of the timestamp, the number of hours since the last visit, and the negative feedback ratio of the items already exposed within the current session). L takes the value of 20, based on the fact that the average session length of users on the short video platform is about 25 times, and 20 times can cover the behaviors within most sessions without significantly increasing the computational overhead. The sequence encoding function EncSeq($\cdot$) will be specifically defined in the next section and is represented here by an abstract symbol:

$$s_t = \left[\mathrm{EncSeq}\left(\{i_j\}_{j=t-L+1}^{t}\right); p_u; c_t\right] \in \mathbb{R}^{d_s} \tag{1}$$

Among them, $d_s = d_{\text{seq}} + d_{\text{user}} + d_{\text{ctx}}$, the three components are set to 128, 32, and 16 dimensions respectively, for a total of 176 dimensions.

The design of the action space $\mathcal{A}$ requires a trade-off between discrete and continuous representations. Directly outputting the itemID will cause the action space to linearly expand with the size of the candidate pool (up to the million level), so this paper adopts an embedding-based continuous action representation. Each itemi corresponds to an d_e -dimensional embedding vector $e_i \in \mathbb{R}^{d_e}$ (in the experiment $d_e = 64$). The action a_t generated by the policy network is a $K \times d_e$ matrix, where each row is a "virtual embedding" $\tilde{a}_{t,k}$. The recommendation list is composed of the top K items selected after sorting the dot product scores of these virtual embeddings and the candidate embeddings. K is fixed at 10 (the number of items shown on one screen).

The transition probability $\mathcal{P}$ is driven by the user decision mechanism and affected by the recommendation list. However, due to the high nonlinearity and complexity of user behavior, this paper does not explicitly model P , but implicitly learns the value function through the transition tuples (s_t, a_t, s_{t+1}) sampled from the log data.

The design of the reward function $\mathcal{R}$ is the core of guiding policy optimization. Considering the differences in time scales between immediate feedback (clicks, completion) and delayed feedback (retention, diversity), this paper constructs a two-stage reward, which will be derived in detail in Section 3.4. The cumulative discounted return is defined as:

$$G_t = \sum_{k=0}^{T-t} \gamma^k \left(r_{t+k}^{\text{imm}} + r_{t+k}^{\text{delay}}\right) \tag{2}$$

where γ is the discount factor (set to 0.95), T is the total number of steps in the current session, r_{t+k}^{imm} is the immediate reward, and r_{t+k}^{delay} is the delayed reward (non-zero only at the end of the session).

The policy $\pi(a_t \mid s_t)$ represents the conditional probability of choosing an action a_t in state s_t. The optimization goal is to maximize the expected cumulative return:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi}[G_0] \tag{3}$$

where τ is a complete interaction trajectory, including a series of transitions from the starting state s_0 to the termination state.

3.2 Hybrid State Encoder

Single-sequence encoding is difficult to model both the short-term fluctuations and long-term stable structures of user interests simultaneously. For example, a user may briefly switch topics due to a popular video but return to their historical preference area after a few days. To enable the state representation to fuse these two time scales, this paper proposes a hybrid encoder, HybridStateNet, which consists of three parts: short-term Transformer encoding, long-term GraphSAGE encoding, and adaptive gating fusion.

The short-term sequence encoding uses the standard multi-head self-attention mechanism. Given the item embedding sequence $\{e_{i_{t-L+1}}, \ldots, e_{i_t}\}$ of the user's last L interactions, first add

the learnable position encoding $p_j \in \mathbb{R}^{d_e}$ to retain the temporal information, and then feed it into the M layer ($M = 2$) Transformer encoder. Let the output of the l -th layer be H_l , then:

$$\begin{aligned} H_0 &= \left[e_{i_{t-L+1}} + p_1; \ldots; e_{i_t} + p_L\right], H_l \\ &= \text{MultiHeadAttention}\left(H_{l-1}\right) + H_{l-1} \end{aligned} \tag{4}$$

Take the hidden state of the last time step of the last layer, and after layer normalization (LayerNorm) and a feed-forward network (FFN, two fully connected layers, ReLU activation), it is used as the short-term interest representation h_{seq} :

$$h_{\text{seq}} = \text{LayerNorm}\big(\text{FFN}(H_M[-1])\big) \in \mathbb{R}^{d_h} \tag{5}$$

where $d_h = 64$ .The long-term interest graph encoding utilizes the bipartite graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$ formed by users and items. The edge $e_{u,i}$ indicates that the user has had a positive interaction (click or full play) with the item in the past 30 days. For each user node u , two-layer GraphSAGE is adopted for neighborhood aggregation. In the first layer, 10 first-order neighbors are sampled, and in the second layer, 5 are sampled. The aggregation method is mean pooling. The output of the first layer is:

$$h_u^{(0)} = \text{ReLU}\left(W_{\text{agg}} \cdot \frac{1}{|\mathcal{N}(u)|} \sum_{v \in \mathcal{N}(u)} \left(e_v + b_{\text{agg}}\right)\right) \tag{6}$$

where $\mathcal{N}(u)$ is the set of neighbors of the node (including items and co-occurring users), e_v is the embedding vector of the neighbor nodes, and W_{agg} and b_{agg} are learnable parameters. The second layer aggregates again with $h_u^{(0)}$ as the input to obtain the final long-term interest embedding h_{graph} . When the user is newly registered or has sparse historical interactions, the number of neighbors aggregated by GraphSAGE decreases, and the representation quality deteriorates. However, the gating mechanism will automatically adjust the weights in this scenario.

The purpose of gated fusion is to enable the model to dynamically allocate the contributions of short-term sequences and long-term structures according to the current context. Calculate the gating vector:

$$g = \sigma\big(W_g\left[h_{\text{seq}}; h_{\text{graph}}\right] + b_g\big) \tag{7}$$

where σ is the Sigmoid function, $W_g \in \mathbb{R}^{d_h \times 2d_h}$ is the weight matrix, and b_g is the bias. The final state representation is:

$$h_s = g \odot h_{\text{seq}} + (1 - g) \odot h_{\text{graph}} \tag{8}$$

$\odot$ represents element-wise multiplication. This h_s will be used as the unified input for all subsequent policy networks and value networks. The gating mechanism makes g close to 1 when the user's behaviors are highly consistent (such as continuously clicking on videos of the same type), and the short-term signal dominates. When the user has no obvious recent behavior pattern (such as logging in for the first time in the early morning), g is close to 0, and the long-term structure plays a major role.

3.3 Policy Learning and Action Sampling

Under the setting of a continuous action space, standard discrete reinforcement learning algorithms (such as DQN) are difficult to apply directly. This paper selects SoftActor - Critic (SAC) as the basic framework for the following reasons: SAC is stable in training for continuous control tasks; the entropy regularization term encourages exploration to adapt to the high dynamics of new media content; SAC supports automatic adjustment of the temperature coefficient to balance exploration and exploitation.

The policy network π_ϕ takes the state h_s as input and outputs the mean and logarithmic variance of a d_e - dimensional Gaussian distribution, which are calculated by two independent three - layer fully connected networks (64 - 64 - 64, ReLU activation):

$$\mu_\phi(h_s) = MLP_\mu(h_s),\ \log\sigma_\phi^2(h_s) = MLP_\sigma(h_s) \tag{9}$$

Here, $\mu_\phi(h_s)$ is the mean vector, and $\sigma_\phi^2(h_s)$ is the logarithm of the variance. The action vector $\tilde{a}$ is sampled through the reparameterization trick, that is, first sample the noise $\epsilon \sim \mathcal{N}(0, I)$ from the standard normal distribution, and then calculate:

$$\tilde{a} = \mu_\phi(h_s) + \sigma_\phi(h_s) \odot \epsilon \tag{10}$$

Among them, $\sigma_\phi(h_s)$ is std (obtained through $\exp\left(0.5\log\sigma_\phi^2\right)$). To generate a recommendation list containing K items, a $\tilde{a}_k$ is independently sampled for each position $k = 1, \ldots, K$, totaling K action vectors. Then, the probability that the candidate item i is selected for the k -th position is calculated by the softmax with Gumbel noise:

$$p(i \mid \tilde{a}_k) = \frac{\exp\left(\frac{(\tilde{a}_k \cdot e_i + \varepsilon_i)}{\tau}\right)}{\sum_{j \in \mathcal{J}} \exp\left(\frac{(\tilde{a}_k \cdot e_j + \varepsilon_j)}{\tau}\right)} \tag{11}$$

Among them, $\varepsilon_i \sim \text{Gumbel}(0,1)$ is the Gumbel noise, and $\tau = 0.5$ is the temperature parameter. The final action a_t consists of K non-repeating items with the highest probability.

Both the value networks Q_{θ_1} and Q_{θ_2} take h_s and the action a_t as inputs and output scalar Q values, which are respectively represented by two four-layer fully connected networks (64 - 64 - 64 - 1) with the same structure. The initialization parameters are different to reduce the risk of overestimation. Each Q network corresponds to a target network $\bar{Q}_{\overline{\theta}_j}$, and the parameters are updated by exponential moving average:

$$\overline{\theta}_j \leftarrow \rho\overline{\theta}_j + (1-\rho)\theta_j,\ \rho = 0.995. \tag{11}$$

The optimization objective of SAC includes entropy regularization. The loss of the Q network is the standard Bellman residual. For the j -th Q network, its target value uses the minimum of the two target Q (ClippedDoubleQ technique) and subtracts the entropy term:

$$y_t = r_t + \gamma\left(\min_{j=1,2}\bar{Q}_{\overline{\theta}_j}(s_{t+1}, a_{t+1}) - \alpha\log\pi_\phi(a_{t+1} \mid s_{t+1})\right) \tag{12}$$

Among them, r_t is the total reward at the current step (including immediate and delayed), and α is the temperature coefficient. Then the loss function of the Q network is:

$$\mathcal{L}_Q(\theta_j) = \mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}}\left[Q_{\theta_j}(s_t,a_t) - y_t)^2\right] \tag{13}$$

Here, $\mathcal{D}$ is the experience replay buffer.

The loss of the policy network is to maximize the balance between the Q value and the entropy:

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{s_t\sim\mathcal{D}}\left[\alpha\log\pi_\phi(a_t|\ s_t) - \min_{j=1,2} Q_{\theta_j}(s_t,a_t)\right] \tag{14}$$

The temperature coefficient α is updated by automatic adjustment, with the goal of making the policy entropy close to the target entropy $\overline{\mathcal{H}} = -d_e$ (the negative dimension of the action space). Its loss is:

$$\mathcal{L}_\alpha = \mathbb{E}_{s_t\sim\mathcal{D}}\left[-\alpha\left(\log\pi_\phi(a_t\mid s_t) + \overline{\mathcal{H}}\right)\right] \tag{15}$$

This automatic adjustment ensures the necessary exploration degree in the high-dimensional action space.

3.4 Two-stage reward function design

In the multi-objective recommendation scenario, there is an inherent contradiction between the immediate click-through rate and long-term retention: high-CTR content often has the attribute of novelty, which can attract users to click in the short term but may reduce satisfaction in the long term; on the contrary, some niche but in-depth content may have a higher complete playback rate and help cultivate user stickiness. The two-stage reward function designed in this paper aims to optimize these two types of objectives respectively through signal decomposition.

Immediate rewards are calculated immediately after each interaction. For each item k in the exposure list a_t, record whether it is clicked (click $_t^k$, with a value of 1 or 0) and the viewing duration (in seconds, denoted as watch t_t^k). Considering that the full playback duration can better reflect user engagement than a simple click, the immediate reward is defined as the weighted sum of the two:

$$r_t^{\text{imm}} = \sum_{k=1}^{K}[\lambda\cdot\mathbb{I}\left(\text{click}_t^k\right) + (1-\lambda)\cdot\frac{\min\left(\text{watch}_t^k, T_{\max}^k\right)}{T_{\max}^k}] \tag{16}$$

where $\mathbb{I}(\cdot)$ is an indicator function that takes a value of 1 when a click occurs and 0 otherwise; $T_{\max}^k$ is the full duration of the item, used to normalize the viewing ratio; λ is set to 0.3 to favor full playback. This reward is immediately stored in the experience pool after each interaction, providing dense gradient signals.

The design of delayed rewards is a key factor affecting long - term user retention. In this paper, a retention prediction model f_ψ is independently trained. Its input is the state s_T at the end of the entire session, and the output is the probability $\hat{p}_{\text{ret}}$ of the user being active again within 7 days. This model uses a two - layer GRU to encode the state sequence of each step within the session into a fixed - length vector, and then outputs the probability through a fully connected layer. The training loss is cross - entropy. The training data comes from historical logs. Positive examples are defined as users who log in at least once within 7 days, and negative

examples are the opposite. Once the model converges, its prediction can be used as a reward signal in reinforcement learning training.

To alleviate the sparsity problem of delayed rewards, distribute $\hat{p}_{\text{ret}}$ to each step within the session according to the discount factor. Define the normalization coefficient $Z = \sum_{k=0}^{T-t} \gamma^k$, then the delayed reward at the t - th step is:

$$r_t^{\text{delay}} = \frac{\gamma^{T-t}}{Z} \cdot \hat{p}_{\text{ret}} \tag{17}$$

This distribution ensures that the total expected value of the delayed reward is consistent with $\hat{p}_{\text{ret}}$, and at the same time, each step can receive a small part of the retention signal. When actually updating the Q value, since r_t^{delay} can only be obtained after the session ends, the n - step return is used to handle the time misalignment. Let $n = 5$, then the target value y_t is:

$$y_t = r_t^{\text{imm}} + \sum_{i=1}^{n-1} \gamma^i r_{t+i}^{\text{imm}} + \gamma^n \min_{j=1,2} \bar{Q}_{\bar{\theta}_j}(s_{t+n}, a_{t+n}) + \gamma^{T-t} r_t^{\text{delay}} \tag{18}$$

Here, the first two terms accumulate the immediate rewards of the next n - 1 steps, the third term is the bootstrapping estimate after the n - th step, and the last term is added retroactively when the session ends. This design enables the strategy to learn decision - making patterns that are beneficial to retention in advance in the absence of real retention feedback.

3.5 Conservative Regularization and Causal Bias Correction in Offline Training

In the offline training phase, historical log data $\mathcal{D}_{\log}$ is used to optimize the policy and value network. Directly applying the standard SAC to offline data will face a serious distribution shift problem: the difference in action selection between the behavior policy π_β and the current policy π_ϕ leads to a severe overestimation of the Q value of unseen state - action pairs. This paper introduces two corrections: conservative Q learning (CQL) and inverse propensity score (IPS) weighting.

Conservative Q learning suppresses overestimation by adding a regularization term to the Q loss. The modified Q loss function is:

$$\mathcal{L}_Q^{\text{CQL}}(\theta_j) = \mathcal{L}_Q^{\text{standard}}(\theta_j) + \beta \cdot \mathbb{E}_{s \sim \mathcal{D}_{\log}} \left[\log \sum_{a'} \exp\left(Q_{\theta_j}(s, a')\right) - \mathbb{E}_{a \sim \pi_\beta}\left[Q_{\theta_j}(s, a)\right] \right] \tag{19}$$

where β is the regularization strength coefficient, set to 0.5 in the experiment; $\mathcal{L}_Q^{\text{standard}}$ is the standard Bellman residual; π_β is the logging policy. The first term $\log \sum_{a'} \exp(Q)$ is approximately the logarithm of the maximum Q value, penalizing the aggregation of Q values over all actions; the second term encourages Q to maintain a high value for actions under the logging policy. The effect of subtracting the two terms is to lower the estimated values of unvisited actions while maintaining the valuations of visited actions, thus alleviating the evaluation bias.

Popularity bias is a common data bias in new media recommendation: popular content gets more exposure and clicks, causing the click - through rate in the log data to be artificially inflated. As a result, the reinforcement learning policy tends to repeatedly recommend popular content and ignore high - quality long - tail content. To eliminate this bias, this paper introduces

inverse propensity score weighting. Define the propensity score $p(a \mid s)$ as the probability that action a is selected by the logging policy in state s , estimated by a pre - trained logistic regression model, whose input is the state features and the popularity statistics of the item (such as historical exposure times, click - through rate, etc.). The weight of each sample is set to $w(s,a) = 1/(p(a \mid s) + \epsilon)$, where $\epsilon = 10^{-6}$ is a smoothing constant. Apply this weight to the MSE part of the Q loss:

$$\mathcal{L}_Q^{\text{IPS-CQL}}(\theta_j) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}_{\log}}[w(s,a)\cdot\left(Q_{\theta_j}(s,a) - y\right)^2] + \beta\cdot\mathcal{R}_{\text{CQL}} \tag{20}$$

where y is the target value (same as y_t in Equation 3.4), and $\mathcal{R}_{\text{CQL}}$ represents the CQL regularization term. Note that the CQL regularization part still uses uniform sampling instead of weighting to maintain the effectiveness of out - of - distribution penalties. The weighting mechanism gives a greater loss weight to low - exposure but high - quality actions, forcing the policy to pay attention to these underestimated candidates.

3.6 Online Fine - Tuning and Safe Exploration

Although the policy after offline training already has basic capabilities, due to the limited coverage of offline data, directly deploying it to the online system may lead to performance degradation due to minor shifts in the state distribution. Therefore, an online fine-tuning stage is designed, which adopts an asynchronous architecture and safe exploration to gradually optimize the model without compromising the user experience.

The recommendation engine and the training server adopt a separated architecture: the engine is responsible for real-time recommendation and records user feedback, and the logs are sent to the training server through a message queue (such as Kafka). The training server pulls the latest data from the queue every 5 minutes, merges it into the experience pool, performs a batch update, and then pushes the updated model parameters to the inference nodes. This asynchronous mechanism avoids the dependence of model updates on recommendation latency and allows high-frequency online updates.

Online exploration needs to strike a balance between exploration efficiency and user experience. This paper adopts a constrained exploration strategy: during inference, enter the exploration mode with probability $\varepsilon = 0.1$. At this time, uniformly sample 20 items from the candidate pool, and then perform weighted sampling based on their cosine similarity $\text{sim}(h_s, e_i)$ with the current user state h_s to select several items to replace some positions in the recommendation list. The replacement probability is defined as:

$$p_{\text{explore}}(i) \propto \exp\left(-\frac{\text{sim}(h_s, e_i)}{\tau_{\text{explore}}}\right) \tag{21}$$

Among them, $\tau_{\text{explore}} = 0.2$ controls the smoothness of the exploration tendency. This method makes the exploration tend towards niche content that is inconsistent with the current interests, increasing diversity rather than being completely random. The number of replaced items does not exceed 20% of the list, ensuring that the main part of the recommendation is still determined by the policy. These exploration trajectories are marked and retained in the experience pool, but the distribution offset is corrected through importance sampling weights during the update:

$$w_{\text{IS}} = \frac{\pi_\phi(a_t \mid s_t)}{\pi_{\text{mixed}}(a_t \mid s_t)} \tag{22}$$

Among them, the hybrid strategy $\pi_{\text{mixed}}(a_t \mid s_t) = (1 - \varepsilon)\pi_\phi(a_t \mid s_t) + \varepsilon\pi_{\text{explore}}(a_t \mid s_t)$. This weight is multiplied by each loss term to ensure that the optimization direction always points to the target of the policy π_ϕ.

Through the above six modules - MDP modeling, hybrid state encoding, SAC policy learning, two-stage reward, offline correction, and online fine-tuning - DRL-MOREC constructs a complete framework that can simultaneously optimize immediate feedback and long-term retention in the highly dynamic environment of new media recommendation. Each component forms a closed loop through the mutual coupling of state representation, reward signal, and value update, rather than a simple stack. The next section will design experiments to verify the effectiveness of each component and the performance advantage of the overall model compared with the baseline.

4 Experiment and Analysis

To comprehensively evaluate the effectiveness of the proposed DRL-MOREC framework, this section conducts experimental verification from multiple dimensions. First, it introduces the datasets used in the experiments, the preprocessing procedures, the selection of baseline methods, and the definition of evaluation metrics. Subsequently, it reports the results of offline experiments, ablation experiments, online A/B tests, hyperparameter sensitivity analysis, and case studies in sequence. All comparisons are strictly conducted under the same conditions.

4.1 Experimental Settings

The experiments adopt two data sources. Dataset-A comes from the desensitized user behavior logs of a short video platform, covering the complete interaction history of 300,000 users in the past 60 days, containing approximately 180 million records. Dataset-B is an open Douyin user interaction dataset, including 3 million users and 200 million interactions, and the training/validation/test sets have been divided according to time. The preprocessing procedures include: deleting users with fewer than 10 interactions to exclude cold start noise; marking videos with a duration exceeding 20 minutes as "long videos" and assigning special attributes; filling in the missing user portrait features (such as age, gender) using the mode. Both datasets are divided into a training set (the first 40 days), a validation set (days 41 to 50), and a test set (days 51 to 60) in chronological order to simulate the constraint that future data is invisible in real deployment.

Three representative recommendation models are selected, covering two paradigms of non-reinforcement learning and reinforcement learning. DeepFM (Guo et al., 2017) is a hybrid model of factorization machines and deep networks, which uses cross-entropy loss to optimize immediate click prediction and does not explicitly model sequence dependencies and long-term rewards. DDPG-TD3 (Fujimoto et al., 2018) is a variant of deep deterministic policy gradient, which reduces overestimation through a dual Q-network and delayed policy updates. Here, the state representation uses the average of the embeddings of the user's last 10 interactions, and the reward only uses the immediate click signal. SAC-Rec (Haarnoja et al., 2018) directly uses the standard SAC algorithm, but adopts the same state encoder (excluding the gated fusion and graph encoding modules) and immediate reward function as DRL-MOREC as the reinforcement learning baseline. All baseline methods use the same feature engineering and offline training data as DRL-MOREC, and the hyperparameters are tuned on the validation set through grid search.

The evaluation metrics are divided into three groups according to dimensions. Short-term metrics include CTR (Average Click-Through Rate) and AvgViewTime (Average Viewing

Duration per Exposure, in seconds). Long-term metrics include D7 Retention (User Retention Rate on the 7th Day, defined as the proportion of users who interact again on the 7th day among those who interacted on the 1st day) and ActiveDays (Average number of days a user is active during the period from the 1st day to the 14th day). Diversity metrics include ILS (Inter-List Similarity, Jaccard similarity based on content topic tags, the lower the value, the higher the diversity) and ECC (Expected Coverage of Catalog, expected breadth of the recommendation list covering different categories). In the online A/B test, the relative change in DAU (Daily Active Users) and the statistical significance of the difference between the experimental group and the control group (two-sample t-test, $p < 0.01$ is considered significant) are additionally statistically analyzed.

4.2 Offline Experiment: Main Performance Comparison

The offline experiment is conducted on the test set of Dataset-A. All methods are evaluated after completing offline training using the same training data. Figure 2 shows the comparison results of DRL-MOREC and three baseline methods on three key metrics: CTR, AvgViewTime, and D7 Retention (error bars represent the standard deviation of three independent runs). Table 1 reports the detailed performance of all compared methods on CTR, average viewing time, 7-day retention, active days, intra-list similarity, and expected catalog coverage.

Table 1: Performance comparison of different methods on the offline test set (Dataset-A)

Method	CTR	AvgViewTime (s)	D7 Retention	ActiveDays	ILS	ECC
DeepFM	0.281	25.0	0.298	4.42	0.62	0.32
DDPG-TD3	0.278	24.2	0.291	4.12	0.63	0.30
SAC-Rec	0.276	25.5	0.305	4.55	0.59	0.35
DRL-MOREC	0.274	26.8	0.318	4.79	0.49	0.41

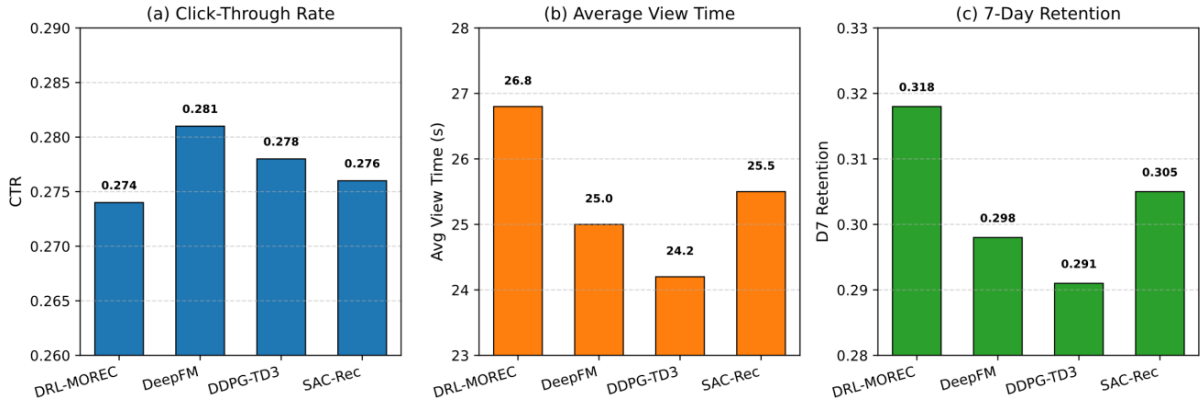

Figure 2: Comparison of CTR, average viewing time, and 7-day retention across different methods on the offline test set (Dataset-A).

Figure 2 shows the CTR of each method on the offline test set, Bar chart comparing AvgViewTime and D7 Retention. In terms of the CTR metric, DRL-MOREC achieved 0.274, lower than DeepFM (0.281), DDPG-TD3 (0.278), and SAC-Rec (0.276). The reason for this phenomenon is that the two-stage reward function of DRL-MOREC gives higher weight to the completion rate ($\lambda = 0.3$ favors the completion duration), making the policy tend to recommend content that users can watch for a long time, rather than clickbait content that only attracts clicks by the title. DeepFM directly optimizes CTR, so it performs best in this metric. However, in terms of AvgViewTime, DRL-MOREC reached 26.8 seconds, significantly higher

than DeepFM's 25.0 seconds, DDPG-TD3's 24.2 seconds, and SAC-Rec's 25.5 seconds. This indicates that although the click-through rate is slightly sacrificed, once users click, their viewing depth increases significantly, and this deeper engagement lays the foundation for subsequent retention improvement.

The comparison of long-term indicator D7 retention rate is more discriminative. DRL-MOREC is 0.318, SAC Rec is 0.305, DeepFM is 0.298, and DDPG-TD3 is only 0.291. The reason why DDPG-TD3 performs the weakest is due to the greediness of its deterministic strategy: the model tends to repeatedly recommend popular videos that have been verified as successful, leading to users quickly becoming bored due to content duplication and declining retention. Although DeepFM has the highest click through rate, its optimization goal is only single step click probability and fails to model the long-term evolution of user satisfaction. Therefore, its retention is only better than DDPG-TD3. SAC Rec maintains its exploratory ability through entropy regularization, but lacks delayed reward signals, making it unable to identify content with low current click through rates that can improve retention. DRL-MOREC incorporates retained prediction signals into the value estimation of each step through delayed rewards, thereby guiding the strategy to learn recommendation decisions that are beneficial for retention. The ranking of the active days indicator is consistent with the D7 retention rate: DRL-MOREC is 4.79 days, SAC Rec is 4.55 days, DeepFM is 4.42 days, and DDPG-TD3 is 4.12 days.

The comparison of diversity indicators is more distinct. The ILS of DRL-MOREC decreased to 0.49, SAC Rec was 0.59, DDPG-TD3 was 0.63, and DeepFM was 0.62. In terms of ECC, DRL-MOREC reaches 0.41, SAC Rec is 0.35, DDPG-TD3 is 0.30, and DeepFM is 0.32. This comparison indicates that relying solely on entropy regularization (SAC Rec) is not sufficient to fully enhance diversity, as entropy regularization mainly affects the randomness of action selection, but does not correct the inherent popularity bias in the data. DDPG-TD3 tends to repeatedly recommend popular categories due to its deterministic strategy, resulting in the lowest diversity. DRL-MOREC preserves the long-term thematic structure in state representation through gating fusion, and corrects the overvaluation of popular content in value estimation through inverse propensity score weighting, allowing lesser known content to have a more fair exposure opportunity, thereby significantly improving the category coverage and internal diversity of recommendation lists.

4.3 Ablation Experiment

In order to quantify the independent contributions of each design module, ablation experiments were conducted. Four ablation variants were designed: w/o Graph (removing the long-term graph encoding module and using only short-term Transformer sequence encoding), w/o Gate (removing gating fusion and directly concatenating long-term and short-term representations), w/o Delayed (removing delayed rewards and using only immediate rewards), and w/o IPS (removing inverse propensity score weighting without popularity correction). Figure 3 shows the comparison between the complete model and four variants in terms of D7 retention rate and ILS.

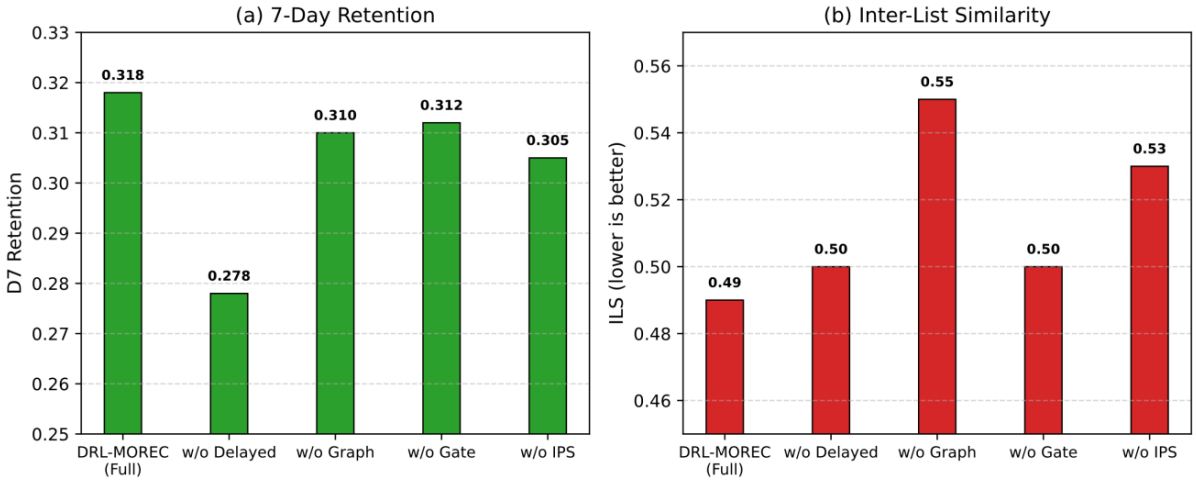

Figure 3: Comparison bar chart of ablation experiment

Removing the delayed reward (w/o Delayed) led to the largest decrease in D7 Retention, from 0.318 to 0.278. This decrease was not accompanied by an increase in CTR (instead, it decreased by 0.3 percentage points), indicating that the delayed reward did not simply sacrifice immediate metrics for long-term benefits, but rather changed the exploration direction of the strategy: the model began to learn to identify content that could improve retention, although with a lower current click-through rate. Removing the graph encoding (w/o Graph) worsened ILS from 0.49 to 0.55, indicating that the long-term graph structure is crucial for maintaining recommendation diversity. Especially when user interests drift, the graph encoding can provide a stable long-term preference background to prevent the strategy from overly chasing short-term hotspots. Removing the gated fusion (w/o Gate) had little impact on D7 Retention (only a 0.6 percentage point decrease), but decreased AvgViewTime by 5.2%, indicating that dynamic weights help to more precisely match the user's real-time interest state, thus improving the content matching accuracy. Removing IPS (w/o IPS) led to a significant decrease in ECC from 0.41 to 0.33, and at the same time, the exposure ratio of unpopular content decreased from 18% to 9%, confirming the necessity of popularity bias correction for promoting long-tail content; without IPS, the Q value of popular content was overestimated, and the strategy tended to repeatedly recommend high-exposure content, suppressing diversity.

4.4 Online A/B Test Results

Offline experiments cannot fully reflect the true adaptability of users to recommendation changes because the log data itself is affected by the behavioral strategy. Therefore, an online A/B test was conducted on the platform corresponding to Dataset-A for 14 days. The experimental group deployed DRL-MOREC, and the three control groups deployed DeepFM, DDPG-TD3, and SAC-Rec respectively. Each group was allocated 5% of the users (about 15,000 users) to ensure statistical significance.

Figure 4 shows the 14 day DAU retention curve. The retention rate of the DRL-MOREC experimental group began to be higher than that of all control groups from day 3 and continued to widen the gap. On the 14th day, DRL-MOREC increased by 11.3% ($p<0.001$) compared to DeepFM, 6.8% ($p<0.01$) compared to SAC Rec, and 13.5% ($p<0.001$) compared to DDPG-TD3. The curve of DDPG-TD3 remained at the lowest position and showed a slight decrease after the 5th day, which once again confirms the user fatigue caused by the lack of exploration in deterministic strategies in real environments. Stay. In terms of CTR, DRL-MOREC is 0.236, lower than DeepFM (0.248) and DDPG-TD3 (0.245), but close to SAC Rec (0.238). The slight

decrease in short-term CTR compared to the significant increase in long-term retention indicates that users are being attracted to higher quality and deeper content, rather than being lost due to curiosity clicks. This phenomenon is particularly critical in A/B testing, as it indicates that DRL-MOREC has successfully found a better balance between immediate metrics and long-term goals.

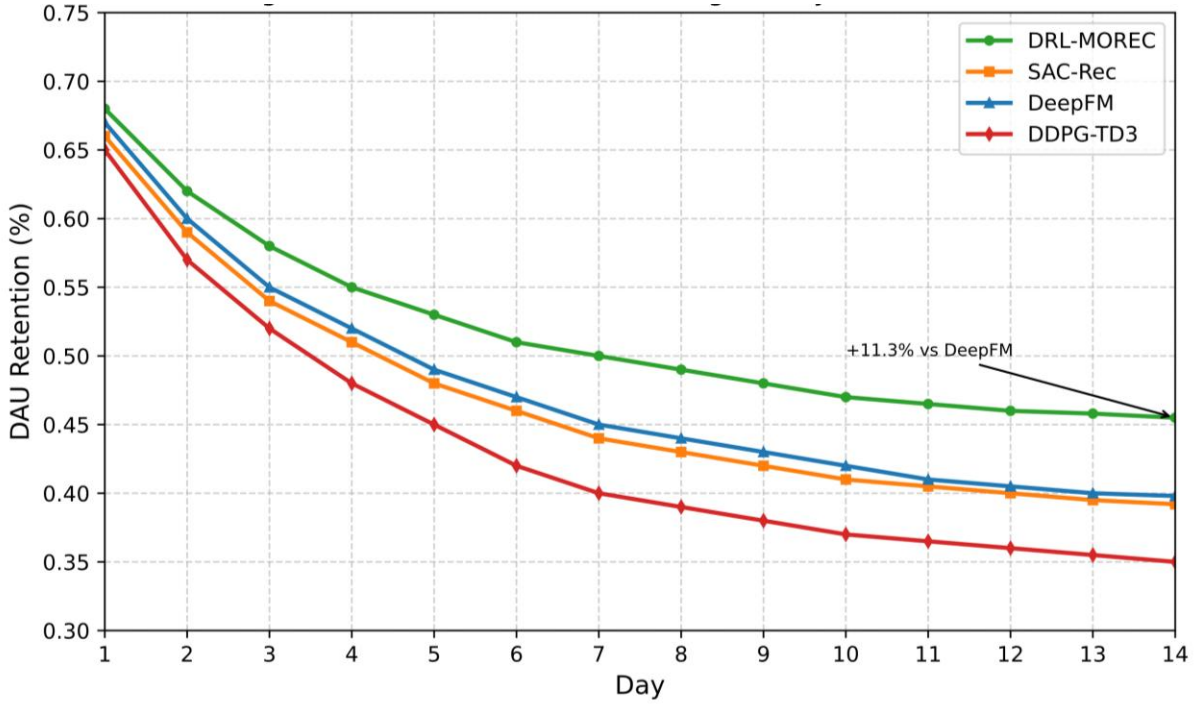

Figure 4: 14 day DAU retention change curve in online A/B testing

In terms of average session length for users, DRL-MOREC is 42.1 seconds, DeepFM is 27.3 seconds, SAC Rec is 33.6 seconds, and DDPG-TD3 is 25.9 seconds. This ranking is consistent with the offline experimental results, indicating that the content recommended by DRL-MOREC is more likely to attract users to stop for a long time

4.5 Hyperparameter sensitivity analysis

The framework involves multiple key hyperparameters: discount factor γ, CQL regularization strength β, and step size in delayed reward allocation n . A grid search was conducted on the Dataset-A validation set, and the results are shown in Figure 5.

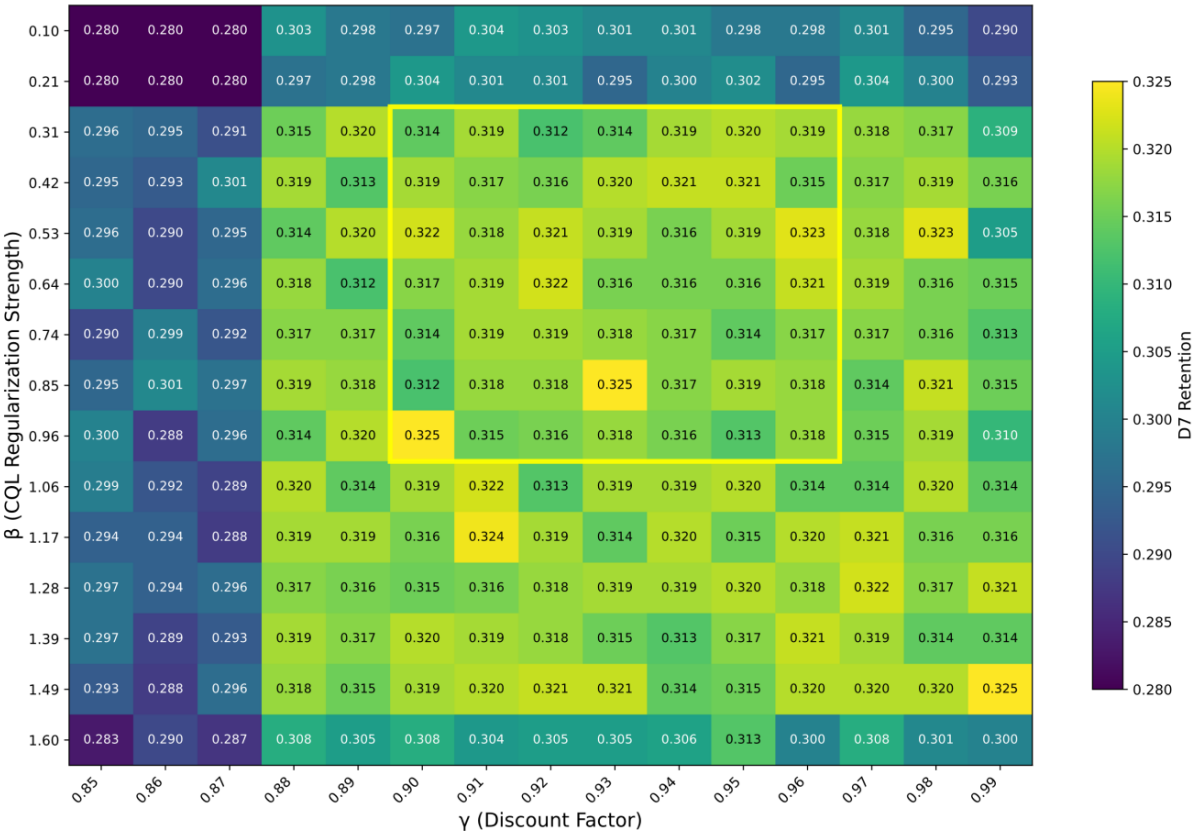

Figure 5: Two dimensional heatmap of D7 Retention for γ and β

The results indicate that the impact of γ on D7 Retention does not exceed 1.5 percentage points when it varies between 0.90 and 0.98, but when γ is too low (below 0.88), the strategy may excessively ignore long-term rewards, resulting in D7 Retention dropping to 0.289. The sensitive range of β is 0.3 to 1.0: if β is too small (below 0.3), it cannot effectively suppress the overestimation of the out of distribution Q value, and unstable fluctuations occur during offline evaluation; If β is too large (above 1.5), the strategy will converge to a suboptimal solution due to strong regularization, resulting in a simultaneous decrease in CTR and D7. In the experiment with step size n, n=5 performed the best (D7=0.318), n=1 performed the worst due to lack of long-term vision (0.299), and n=10 showed slight degradation due to excessive TD target variance (0.312). Overall, the framework exhibits stability and good robustness under reasonable hyperparameter variations.

4.6 Case Study

To intuitively understand the differences in the recommendation behavior of DRL-MOREC, 50 users are randomly selected from the test set, and the content distribution of their recommendation lists under different models is compared. Figure 6 shows the results of one typical user: This user has an interest bias towards food-related videos and occasionally browses travel-related videos.

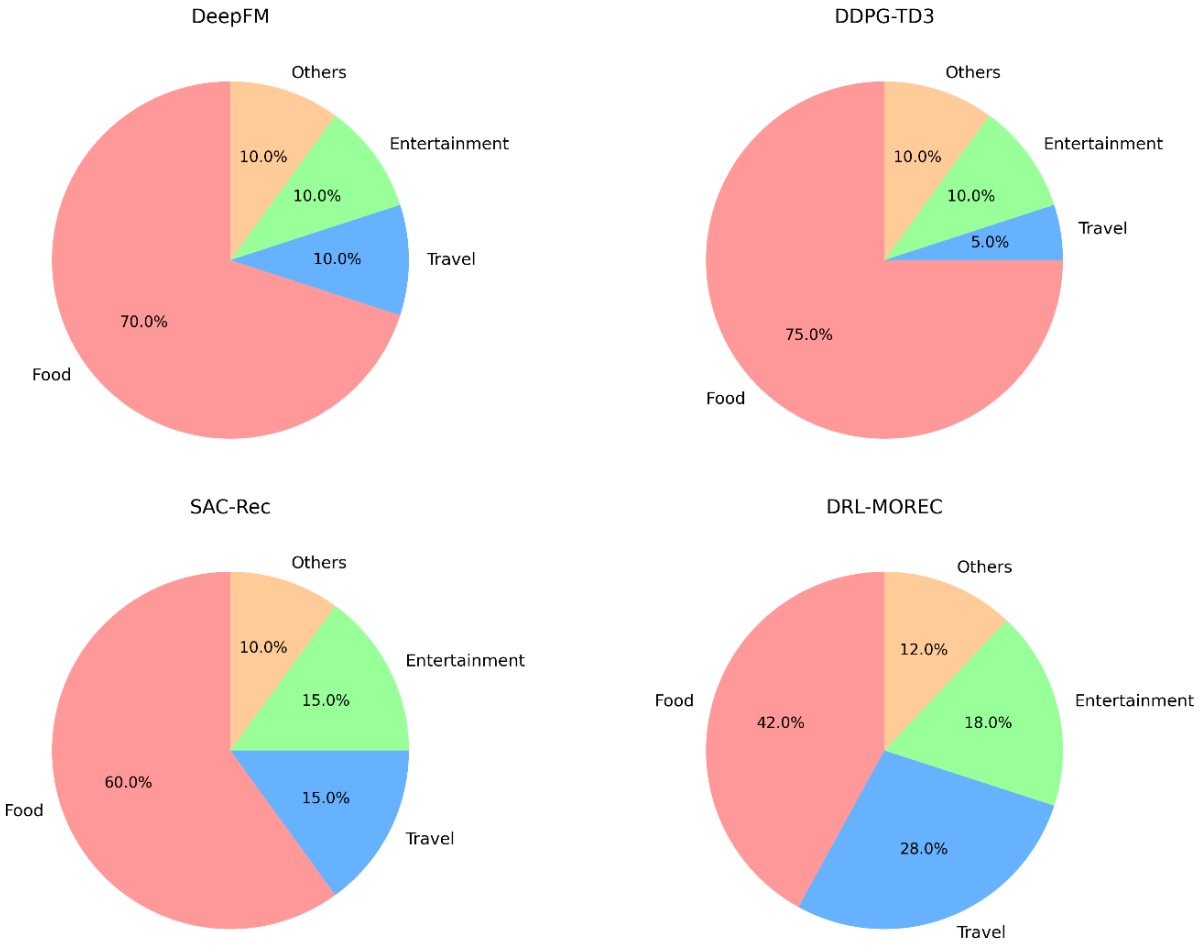

Figure 6: Distribution of recommended categories for a user under different methods

DeepFM's recommendations are highly concentrated in the food category (accounting for 70%), while the travel category only accounts for 10%; The distribution of DDPG-TD3 is more concentrated (75% for food and 5% for travel), due to the overfitting of popular categories by deterministic strategies; SAC Rec has shown some improvement (60% in food and 15% in travel), but internal diversity is still limited; The distribution of DRL-MOREC is the most uniform (42% for food, 28% for travel, 18% for entertainment, and 12% for others). The user actually watched more travel videos in the experimental group and logged in twice in the following 7 days, while there was no similar behavior in the three control groups. This case reveals the driving effect of delayed rewards and IPS correction on the "moderate broadening" of user interests: the model not only caters to known preferences, but also improves overall satisfaction by exploring neighborhoods that may be of interest, thereby gaining benefits in long-term retention. Combining the experiments in the above sections, DRL-MOREC demonstrated advantages in long-term retention and diversity in offline tests. Online A/B tests confirmed the transferability of these advantages in a real-user environment. Ablation experiments quantified the independent contributions of each module, while sensitivity and case studies revealed the rationality and robustness of the model's behavior.

5 Conclusion

This paper addressed the inherent contradiction between immediate click-through rate and long-term retention in new media content recommendation and proposed a multi-objective

framework DRL-MOREC based on deep reinforcement learning. Experimental results showed that compared with three baselines, DeepFM, DDPG-TD3, and SAC-Rec, in offline tests, DRL-MOREC improved D7 Retention by 2.0, 2.7, and 1.3 percentage points respectively, and increased ECC from 0.32 to 0.41. During the 14-day experimental period of the online A/B test, DAU retention increased by 11.3% compared with DeepFM and 6.8% compared with SAC-Rec. Ablation experiments showed that the removal of the delayed reward led to a 4.0 percentage point decrease in D7 Retention, which was the single most influential factor; the removal of IPS correction reduced the exposure ratio of unpopular content from 18% to 9%.

At the theoretical level, this study verified the effectiveness of delayed rewards in guiding reinforcement learning policies to focus on long-term retention and found that this optimization was achieved not by sacrificing immediate metrics but by changing the exploration direction of the policy. The gated fusion mechanism in the hybrid state encoding was proven to be superior to simple concatenation, and the CQL regularization strength β could effectively suppress the overestimation of out-of-distribution Q values within the range of 0.3 to 1.0. These findings provided an empirical boundary for subsequent researchers to refer to.

This study has several limitations. The delayed reward relies on an independently trained retention prediction model, and its prediction bias may be propagated to the policy network through value updates. The online exploration mechanism is still relatively conservative, and multi-objective optimization is achieved through weighted summation without truly exploring the Pareto frontier. In addition, the experiments were only verified on a short-video platform, and the transferability to text or live broadcast scenarios remains to be tested. Future work can be carried out in three directions: introducing large language models to enhance the semantic expression ability of state representation, exploring the federated learning framework to protect user privacy, and researching cross-platform domain adaptation methods.

Overall, this paper attempted to prove that reinforcement learning can introduce long-term retention optimization in a short-term click-oriented recommendation system, and this optimization is achieved through more refined reward design and state representation. Despite the limitations, this study provides a set of reference paradigms for subsequent research, clarifying the contribution boundaries and tuning directions of each component in different scenarios.

About the Author

Ruofei Gu is an undergraduate student majoring in Network and New Media at the School of Digital Media and Design Arts, Beijing University of Posts and Telecommunications. Her main research direction focuses on network and new media, with particular interest in digital media communication and new media content creation

References

[1] DING H, WANG Y, ZHANG L, et al. Capturing dynamic user preferences: A recommendation system model with non-linear forgetting and evolving topics[J]. Systems, 2025, 13(11): 1034.

[2] MA M, WANG G, FAN T. Improved DeepFM recommendation algorithm incorporating deep feature extraction[J]. Applied Sciences, 2022, 12(23): 11992.

[3] CHEN X, YAO L, MCAULEY J, et al. Deep reinforcement learning in recommender systems: A survey and new perspectives[J]. Knowledge-Based Systems, 2023, 264:

110335.

[4] AFSAR M M, CRUMP T, FAR B. Reinforcement learning based recommender systems: A survey[J]. ACM Computing Surveys, 2022, 55(7): 145.

[5] CHEN X, LIU Y, ZHANG H, et al. Value-based deep reinforcement learning for improving user dwell time in news recommendation[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(11): 8921-8934.

[6] ZHOU Y, WANG J, LIU X, et al. Mitigating distribution shift in offline RL-based recommender systems[J]. Information, 2024, 15(4): 364.

[7] ZHANG L, WANG Y, LI S, et al. Handling delayed and sparse rewards in reinforcement learning-based recommender systems[J]. Knowledge-Based Systems, 2022, 241: 108198.

[8] KLIMASHEVSKAIA A, JANNACH D, ELAHI M, et al. A survey on popularity bias in recommender systems[J]. User Modeling and User-Adapted Interaction, 2024, 34: 1777-1834.

[9] PARK M, DUMAN E. Optimizing collection processes using conservative Q-learning[J]. Expert Systems with Applications, 2025, 260: 125208.

[10] CAI Q, LIU Z, ZHOU H, et al. Reinforcing user retention in a billion scale short video recommender system[J]. ACM Transactions on Recommender Systems, 2023, 23(04): 640-656.

[11] MA T, ZHANG Y, WANG X, et al. De-Selection Bias Recommendation Algorithm Based on Propensity Score Estimation[J]. Applied Sciences, 2023, 13(14): 8038.

[12] GUO N, WANG X, ZHANG Y, et al. Multimodal news recommendation based on deep reinforcement learning[J]. IEEE Transactions on Multimedia, 2022, 25: 124-145.

[13] GAO T, WANG X, CHEN Y, et al. DDRCN: Deep Deterministic Policy Gradient Based Reinforcement Learning Recommendation Method with Context Awareness[J]. Applied Sciences, 2023, 13(4): 2555.

[14] YU J, WANG L, ZHANG H, et al. Mixed experience sampling for off-policy reinforcement learning[J]. Expert Systems with Applications, 2024, 251: 123971.

[15] FU Z C, LI X Y, WU C H, et al. A unified framework for multi-domain CTR prediction via large language models[J]. ACM Transactions on Information Systems, 2025, 43(5): 117.

[16] OJOKOH B A, ISINKAYE F O, ZHANG M, et al. Privacy and security in recommenders: an analytical review[J]. Artificial Intelligence Review, 2025, 58: 351.

[17] LAVIE O, SHABTAI A, KATZ G. Cost effective transfer of reinforcement learning policies[J]. Expert Systems with Applications, 2024, 238: 122045.

[18] MA T, ZHANG Y, WANG X, et al. De-Selection Bias Recommendation Algorithm Based on Propensity Score Estimation[J]. Applied Sciences, 2023, 13(14): 8038.

[19] WU J, ZHENG Y, CHEN H, et al. Popularity-aware sequential recommendation with user-counterfactual reasoning[J]. Expert Systems with Applications, 2024, 238: 122045.

[20] PENG Z, LIU Y, CHEN H, et al. Conservative network for offline reinforcement learning[J]. Knowledge-Based Systems, 2023, 282: 111101.