CIT. Journal of Computing and Information Technology, Vol. 34, No. 2, June 2026, 117–132
doi: 10.20532/cit.2026.1006393

A Multi-Agent Reinforcement Learning Framework for Personalized Product Recommendation in Live-Streaming E-Commerce: Integrating Improved QMIX with Graph Attention Networks and Deep Reinforcement Learning

Zihan Yang*[1], Sibo Wang[2], Bingjie Hou[3] and Jianming He[4]

[1]Woosuk University, School of Economics and Management, Longdong University, Qingyang, China
[2]Xi'an Physical Education University, School of Education, Longdong University, Qingyang, China
[3]Nottingham Trent University (UK), School of Economics and Management, Longdong University, Qingyang, China
[4]La Trobe University (Australia), School of Economics and Management, Longdong University, Qingyang, China

With the rapid development of live-streaming e-commerce, achieving accurate and personalized product recommendations in a dynamic, real-time, multi-user environment poses a serious challenge to traditional methods. These methods struggle to effectively integrate global collaborative signals and accurately model the dynamic evolution of user interests. This study proposes a personalized recommendation optimization framework that combines a Q-value mixing network algorithm for multi-agent reinforcement learning (QMIX). The framework introduces a multi-user information fusion module built on a graph attention network (GAT) and recurrent neural network (RNN), which constructs unified team-level state representations from heterogeneous local agent observations. Building on this fused representation, a Soft Actor-Critic (SAC)-based deep reinforcement learning mechanism with dual gate recurrent unit encoders models users' short-term interaction dynamics and long-term preference stability, enabling continuous strategy optimization. Experiments on the Movielens-100K and Amazon datasets demonstrate that the proposed model achieves Accuracy@20 values of 0.2051 and 0.0803, respectively, outperforming strong baselines including OneRec and DSETRec across Accuracy@20, Recall@20, F1@20, and cumulative reward metrics. The proposed framework offers a scalable and effective solution for multi-user collaborative recommendation in live-streaming commerce scenarios.

ACM CCS (2012) Classification:

Information systems → Information systems applications → Decision support systems → Expert systems

Keywords: live-streaming e-commerce, multi-agent reinforcement learning, QMIX, graph attention network, personalized recommendation, deep reinforcement learning

1. Introduction

In recent years, live streaming e-commerce, as an emerging form of e-commerce, has quickly become an important platform for promoting consumption upgrades due to its advantages such as real-time interaction, scene immersion, and vivid product display. However, with the rapid growth of user scale and number of products, how to achieve accurate and personalized product selection recommendations in a complex multi-user and multi-product interactive environment has become a key issue to improve user experience and platform efficiency [1]. Traditional recommendation systems mostly rely on methods such as collaborative filtering, content recommendation or sequence models. Although these methods can mine user preferences to a certain extent, in a dynamic, real-time, high-dimensional live broadcast e-commerce environment, it is often difficult to fully integrate the collaborative relationship and global status information among multiple users, resulting in limited recommendation effects [2]. Q-value mixing network algorithm for multi-agent reinforcement learning (QMIX), as a typical multi-agent reinforcement learning framework, achieves non-linear fusion of global state and local observations by introducing a hybrid network, ensuring the consistency of the team's overall goal and individual decision-making. It has demonstrated excellent performance in collaborative control and multi-task decision-making [3-4].

Based on this, this study innovatively builds a live broadcast e-commerce personalized product selection and recommendation optimization model that integrates multi-user information mining. The new model realizes dynamic fusion and representation extraction of multi-user information through graph attention network (GAT) and recurrent neural network (RNN), and builds unified team status characteristics. The study then introduces a deep reinforcement learning mechanism, combined with user short-term interaction sequences and long-term preference feedback, to achieve dynamic modeling of user interests and continuous optimization of recommendation strategies. This research aims to propose a multi-user information fusion and personalized recommendation optimization framework based on the QMIX algorithm. This can solve the problems of dynamic changes in user interests and low efficiency of multi-user collaborative recommendation in live e-commerce scenarios.

2. Literature Review

With the rapid development of live streaming e-commerce, traditional recommendation methods face significant challenges in coping with dynamic changes in user interests, mining multi-user collaborative relationships, and balancing global-local decision-making. For example, in the current study, Li *et al.* proposed a new method that integrates deep learning models and neural matrix factorization models to automatically mine potential relationships between users. Research results showed that the new method could improve the recommendation recall rate of the recommendation system and reduce the loss function [5]. This showed that the user recommendation effect of the recommendation system could be significantly improved through deep learning models. To improve the personalized recommendation effect, user experience, and product sales, Messaoudi *et al.* proposed a recommendation system based on deep neural collaborative filtering. Research results showed that the new system could recommend personalized products based on user experience and could effectively increase user click-through rates [6]. This showed that the recommendation effect of the recommendation system could be significantly improved through deep learning network models. In view of the increasingly important trend of intelligent decision-making technology in complex multi-agent collaborative tasks, Jin *et al.* conducted a systematic review of current research on multi-agent collaborative decision-making. The research focused on discussing these two types of methods and elaborating on their classification, advantages and disadvantages. Finally, this study further looked into the future research directions and key challenges in the field of multi-agent collaborative decision-making, providing an important reference for the development of this field [7]. Bodapati *et al.* proposed a hybrid recommendation method that combines collaborative filtering, content recommendation and deep learning. The experiment was based on the H&M dataset for multi-modal feature fusion and model construction. The results showed that this method was better than the traditional model in key indicators such as accuracy and recall. This provided effective personalized recommendation solutions for areas such as fashion retail [8].

In personalized recommendations for live e-commerce, deep learning methods and multi-agent methods can significantly improve the recommendation effect. In the study of Din *et al.*, focusing on the key role of deep reinforcement learning in medical tasks such as diagnosis, treatment planning, and personalized care, they systematically analyzed its innovative applications and potential challenges in the fields of cancer detection, drug development, robotic surgery, and personalized recommendation of user products. The research pointed out that although DRL had advantages in processing dynamic high-dimensional data, its application in personalized recommendation of user products still faces practical problems such as scarcity of annotation data. To this end, by critically reviewing existing methods and identifying research gaps, this work provided theoretical support and technical direction for the development of the next generation of trustworthy, efficient, and fair artificial intelligence medical solutions [9]. To predict users' future interaction behaviors more accurately, Wang *et al.* proposed a guided diffusion recommendation framework called LeadRec to address the limitations of existing diffusion models that fail to fully utilize sequence information and lack directionality in the generation process. Through comparative experiments on three public datasets, this study showed that the LeadRec framework shows better recommendation performance than existing methods. This verified its effectiveness in improving personalization effects in sequential recommendation scenarios [10].

In summary, although traditional models can significantly improve recommendation effects, traditional methods lack criticism on the applicability of various models under practical constraints such as real-time, scalability and cold start of live broadcast e-commerce. In addition, they fail to establish the evolutionary logic and complementary relationship framework between methods such as deep learning, multi-agent learning and diffusion models, *etc.*, resulting in traditional research being unable to better integrate and analyze user information. To this end, this study proposes a new model using the QMIX algorithm to improve the user information fusion effect.

3. Research Methodology

3.1. User information fusion method based on improved QMIX algorithm

The QMIX algorithm, because of its structured and goal-oriented fusion mechanism, enables the algorithm to pass through a centralized mixing network. It performs non-linear, state-dependent dynamic fusion of the local observation information and global state information of each agent, rather than simple addition. This fusion method can fully mine global information during the training phase and accurately evaluate the value of the team's joint actions. At the same time, it ensures the consistency of the fused team goals and individual decision-making goals through monotonicity constraints and ensures that the decentralized execution strategy can directly serve the overall optimal [11-12]. Therefore, this study uses the QMIX algorithm and RNN as the main algorithm structure of the information fusion model. The entire agent algorithm is composed of an RNN, and the value function is observed by obtaining user historical information and team information. Figure 1 shows the overall structure of the QMIX algorithm.

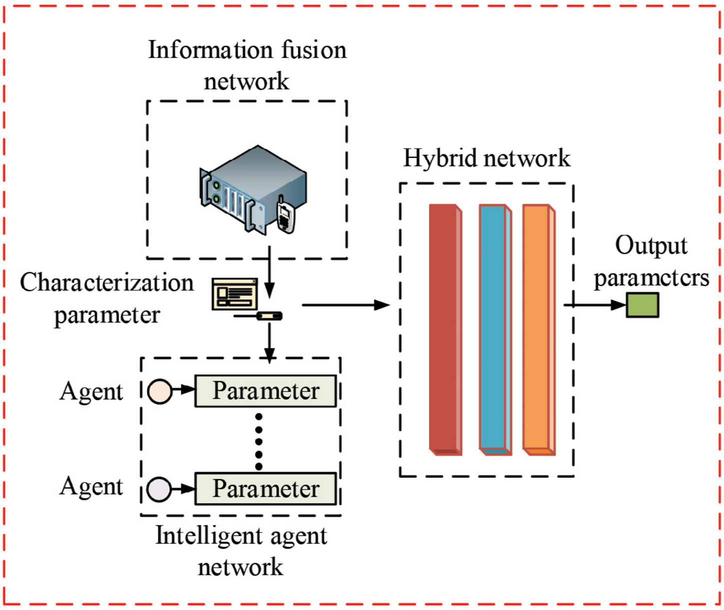


Figure 1. Overall structure of the improved QMIX algorithm.

In Figure 1, the entire algorithm contains three main structures, information fusion network, intelligent agent network and hybrid network. Among them, the information fusion network is mainly responsible for integrating the global information status of all users and outputting unified fusion representation parameters. Therefore, when making decisions, the entire intelligent agent network does not only rely on its own local observation information but uses both local observation and global fusion information as input. This enables each agent's decision-making to be based on a global perspective that integrates all user information. Finally, the hybrid network coordinates and synthesizes the output of each agent to ensure that decentralized decision-making is consistent with the overall optimal goal based on fused information, achieving a complete closed loop from multi-user information fusion to collaborative decision-making. Among them, the intelligent agent network consists of a transformation matrix, underlying parameters, and a tower structure, as shown in Figure 2.

In Figure 2, the intelligent agent network first maps the heterogeneous observation states of different agents to a unified state representation space through the Projection layer transformation matrix. Then, these unified representations are input to the Shared-Bottom layer, allowing all parameter data to learn common features and patterns from all agents, thereby accelerating training and enhancing the overall generalization ability of the model. Finally, the processing flow enters the parallel Tower layer. Each agent has an independent tower network and further learns differentiated and refined strategies that adapt to its own action space and tasks. Finally, it realizes the personalized decision-making of the agent under a unified framework. To ensure that the entire user information can be retrieved and fused in the information fusion network, the agent's attention coefficient is introduced into the model for projection analysis, as shown in Equation (1), which is the attention coefficient formula [13-15].

$$g_{ij}^{h} = s_h^T \bullet [e_i^{'} \| e_j^{'}] \tag{1}$$

In Equation (1), g_{ij}^{h} represents the attention coefficient of the agent node i and the agent node j in the hth attention. s_h^T represents the transpose of the learnable weight vector of the hth attention head. $e_i^{'}$ and $e_j^{'}$ represent the agent i and the agent j of the transformation feature vector. $[e_i^{'} \| e_j^{'}]$ represents the joint feature vector after

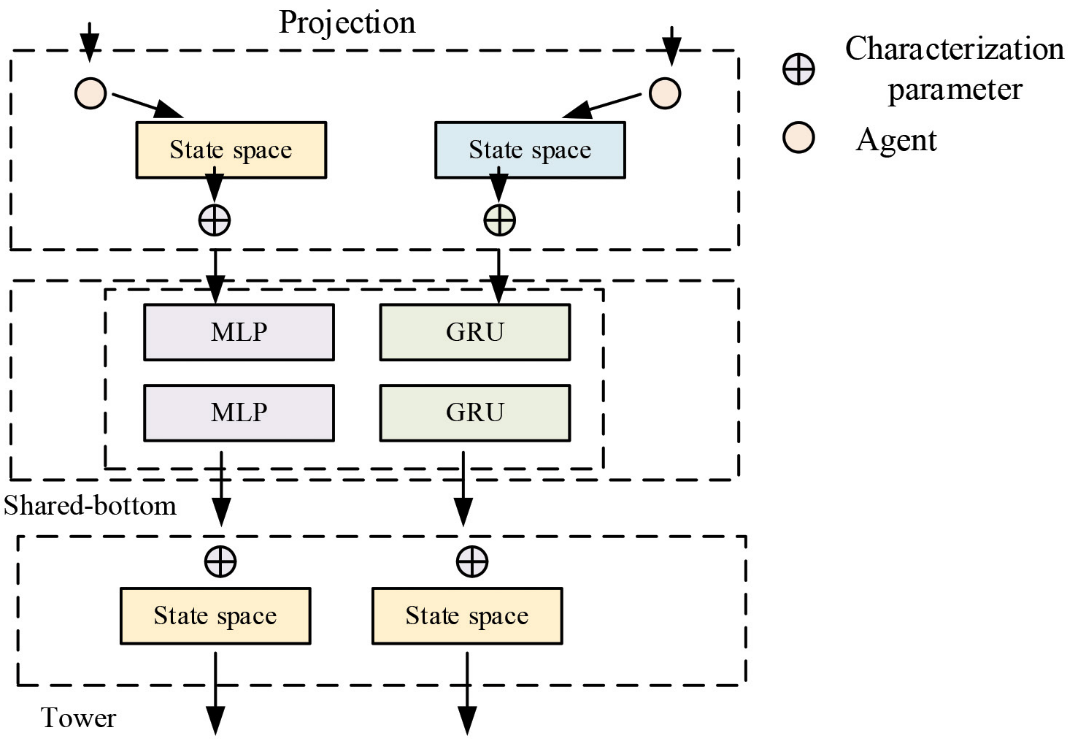


Figure 2. Intelligent agent network.

splicing the feature vectors of two agents. The calculation formula of learning weight vector transpose is shown in Equation (2) [16-17].

$$s_h^T = \text{Softmax}_j(g_{ij}^h) = \frac{\exp(g_{ij}^h)}{\sum_{z \in N_i} \exp(g_{iz}^h)} \tag{2}$$

In Equation (2), $\sum_{z \in N_i} \exp(g_{iz}^h)$ represents the normalized denominator. N_i represents the neighbor set of agent i. z means traversing all neighbors of i. The weighted summation of the neighbor sets of all agents yields the formula shown in Equation (3).

$$s_i^h = \sum_{j \in N_i} g_{ij}^h e_j^{'} \tag{3}$$

After weighted summation, the final eigenvector calculation formula of the feature body is shown in Equation (4) [18-19].

$$c_i = \Big\Vert_{h=1}^{H} s_i^h \tag{4}$$

In Equation (4), c_i represents the final feature vector of agent i after fusing all attention head information. H represents the total amount of attention. The pooling process that combines all user information is shown in Equation (5).

$$e_{all} = \frac{1}{N} \sum_{i=1}^{N} z_i \tag{5}$$

In Equation (5), e_{all} represents the global feature vector of the team that integrates the information of all agents. z_i represents the final feature vector of the ith agent. Fusion and calculation of all the above formulas are as shown in Equation (6).

$$e_{all} = \frac{1}{N} \sum_{i=1}^{N} \left(\Big\Vert_{h=1}^{H} \sum_{j \in N_i} \frac{\exp(g_{ij}^h)}{\sum_{z \in N_i} \exp(g_{iz}^h)} e_j^{'} \right) \tag{6}$$

Equation (6) is the team global feature integration expression of the average pooled features of all fused agent information feature vectors. The final hybrid network monotonically fuses all agents as shown in Equation (7) [20-21].

$$A_{tot} = Mix(A_1(e_1, e_{all}), A_2(e_2, e_{all}), \cdots, A_N(e_N, e_{all}); s_t) \tag{7}$$

In Equation (7), A_{tot} represents the team's joint action function. Mix represents the hybrid network function. $A_N(e_N, e_{all})$ represents the individual action function of the ith agent. e_N represents the local observation of the agent. The fusion structure of the entire hybrid network is shown in Figure 3.

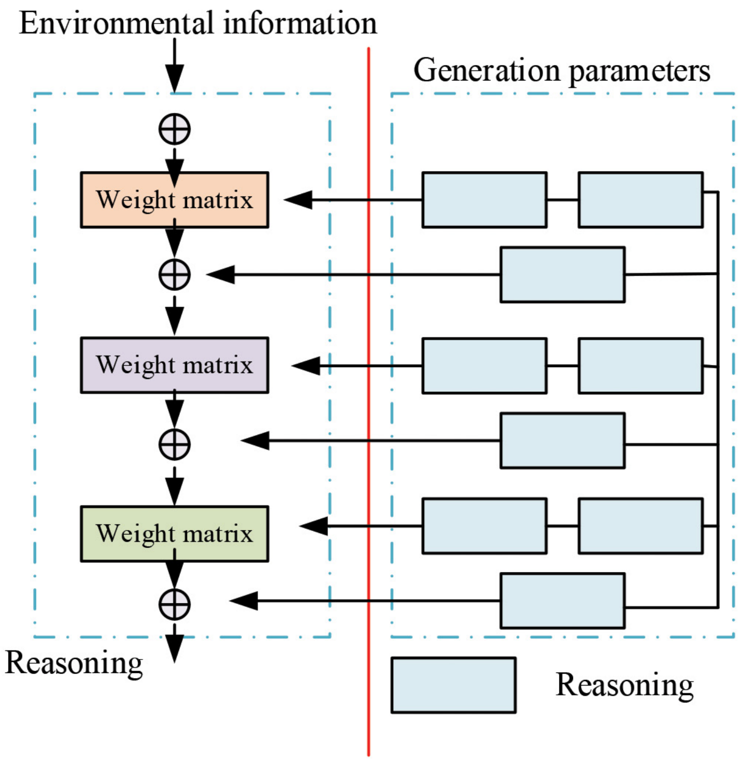


Figure 3. Fusion structure of hybrid network.

As shown in Figure 3, the hybrid network structure contains a parameter generation network and an inference network. Among them, the parameter generation network receives team characteristics and global status that represent global environmental information as input and dynamically generates weight matrices and bias vectors. These generated parameters are then configured on-the-fly to the inference network, which is a feedforward neural network of recurrent networks. The inference network takes the individual values output by each intelligent agent network as direct input, uses the dynamic parameters provided by the parameter generation network to perform forward calculations, and finally outputs a team joint value that satisfies the monotonicity constraint. The hybrid network performs centralized fusion optimization by leveraging global information during the training phase while ensuring deployment execution time. The end user information fusion network is shown in Figure 4.

In Figure 4, the system first obtains all agent user information from the environment for joint observation. Then, the graph is processed through the GAT, the user information of each agent is integrated, and a unified team global feature is generated. On this basis, each agent uses an RNN to calculate its individual value based on its own local observations, action history and global characteristics. Each agent determines its actions independently by running the strategy. After all agents perform joint actions, the environment returns to the team reward and next moment state. Finally, the system stores the observation, status, action, reward, next status, graph structure and other information at the current moment as a piece of experience data into the experience playback buffer.

3.2. Optimization model for personalized product selection and recommendation for live broadcast e-commerce based on user information

After the QMIX algorithm model completes the fusion of user information, it perceives the user's status through recommended information, analyzes the user's interests and hobbies based on the user's perceived changes, and then implements the user's personalized product recommendations. To this end, this study uses deep reinforcement learning to implement personalized product selection recommendations for live streaming e-commerce users based on user fusion information. The intelligent recommendation optimization system uses the user's

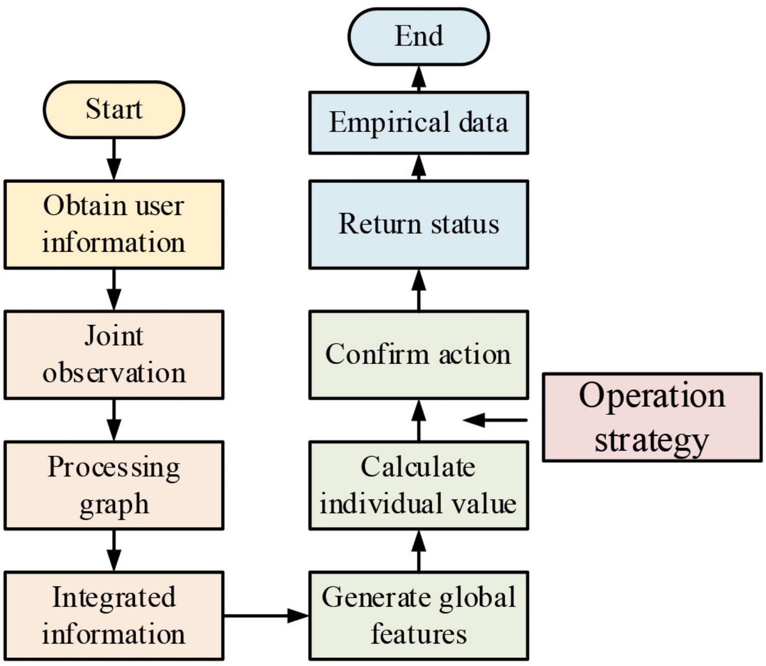


Figure 4. End user information fusion network.

current browsing data as a representation of his recent status and then infers the user's short-term interest preferences by analyzing this status. Among them, the agent captures user information changes through the gated loop unit of reinforcement learning. As shown in Equation (8), it is the update gate state formula [22-23].

$$b_n = \sigma(W_b d_n + U_b k_{n-1}) \tag{8}$$

In Equation (8), n represents the current moment. b_n means update gate. σ represents activation function. W_b represents the update gate input weight matrix. d_n represents the input vector. U_b represents the cyclic update gate weight matrix. k_{n-1} stands for forgotten memory. The reset gate calculation method is shown in Equation (9).

$$r_n = \sigma(W_r d_n + U_r k_{n-1}) \tag{9}$$

In Equation (9), r_n represents the reset gate. W_r represents the reset gate input weight matrix. U_r represents the update gate cycle weight matrix. The gated forgotten memory formula is shown in Equation (10) [24-25].

$$\begin{cases} k_n^{'} = tahn(Wd_n + U(r_n \bullet k_{n-1})) \\ k_n = (1-b_n)k_{n-1} + b_n k_n^{'} \end{cases} \tag{10}$$

In Equation (10), *tahn* represents the candidate hidden state. New encoders are introduced during model training. The main purpose of the encoder is to ensure that the model can effectively capture sparse but critical user high-scoring feedback behaviors, avoid the instability of reinforcement learning model training caused by frequent fluctuations in short-term interests, and thereby achieve the memory and update of users' long-term, stable preferences. Figure 5 shows the final recommendation model framework.

In Figure 5, the gated recurrent unit (GRU) module in the deep reinforcement learning model is responsible for processing the recent interaction sequence between the user and the product, capturing and outputting the state that represents the user's short-term interest. It is then input to the MLP network, which is used to generate recommended actions and estimate their individual values. The system obtains feedback from users after performing recommendations and inputs these feedback sequences into an independent GRU module on the right. The output of the GRU is then aggregated through the average pooling layer to finally form a stable state representation that can

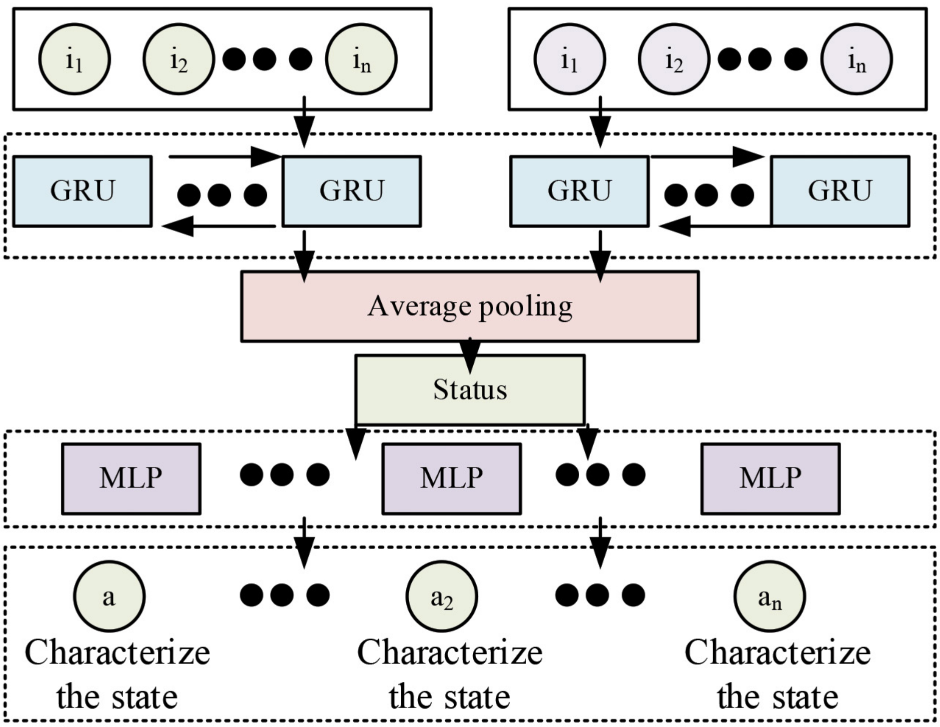


Figure 5. The final recommendation model framework.

reflect the user's long-term preferences. Meanwhile, to ensure that the model can be more stable throughout the process, algorithm training through random strategies is studied. This training method can improve the stability of the model and consider the best user information search in all model states [26]. Figure 6 shows the model training process.

In Figure 6, during the entire model training, the system will first initialize, create an experience replay buffer pool, randomly set the parameters in the SAC algorithm, and then randomly select users. A product that sets its initial state and gets feedback from users watching the item. Then, observations are made through the model, and user status actions are obtained based on the policy network. According to the predetermined matching rules, the system selects the most suitable products from the set of candidate products for recommendation, and at the same time obtains user feedback information in real time. The user status is updated according to the Markov decision process through product recommendation feedback, and the interactive experience tuples are stored in the buffer pool. Then small batches of training data are sampled from the buffer pool, and the parameters of each network in SAC are updated by optimizing the corresponding value function and policy loss respectively. If the current user's interaction sequence has not terminated, return to the observation state step and continue execution. Otherwise, return to the user selection step to process the next user, and loop until all users complete training. The entire recommendation optimization process first fuses user information by improving the QMIX algorithm. Then, the deep reinforcement learning model is used to retrieve and analyze user information, thereby realizing the optimization of personalized product selection and recommendation for live e-commerce based on user information mining. Compared to the standard QMIX algorithm, this study mainly achieves two core improvements: First, a global attention mechanism (GAT) is introduced into the decision input of the agents to construct a multi-user information fusion module, and global user state features are used as auxiliary inputs, enabling the distributed decision-making of each agent to perceive overall collaborative information. Second, the hybrid network is transformed from a static structure to a dynamic parameter generation network, using the team state to generate weights and biases in real time, enabling the joint Q-value estimation to adaptively adjust with environmental changes. In the global embedding dimension, group

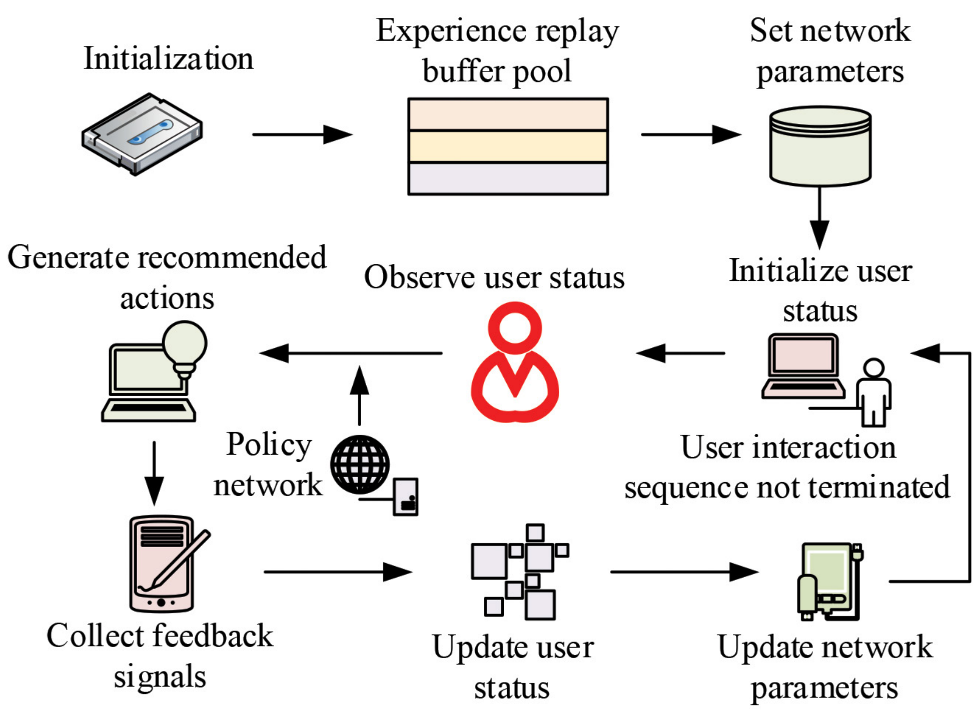


Figure 6. Model training process.

information is extracted from the interaction data of all users through GATs, generating global features as auxiliary inputs for each agent. In the new training dimension, a dynamic parameter generation network is introduced to output the weights and biases of the hybrid network based on real-time team status, and incorporate graph structure information into experience replay, enabling the training process to adapt to dynamic changes in interactive contexts. In the new objective and new constraint dimensions, the Bellman error minimization objective and monotonicity constraint of standard QMIX are used. Overall, the modification list focuses on data fusion and dynamic adaptability enhancement at the architecture layer, rather than refactoring the learning task itself.

The research interprets the reinforcement learning framework as a hierarchical architecture: each agent in the system corresponds to a user, and all users share the environment and make independent decisions. The observation of each agent is their personal historical interaction sequence and short-term interest representation, while the state contains the global team characteristics fused by the GAT. The action is to select recommended products from the candidate set, the reward is user feedback, and the termination condition is to reach the preset interaction steps or user exit. In this framework, the QMIX module serves as the multi-agent coordination layer, fusing the individual Q-values of all users through a hybrid network and ensuring monotonicity constraints, while the SAC algorithm serves as the internal policy learner for each user agent, responsible for generating personalized recommendation policies. The two are jointly optimized during centralized training, sharing an experience replay buffer containing graph structure information, thus achieving the unity of multi-agent collaboration and individual policy learning.

4. Results and Discussion

4.1. Analysis of the effects of personalized product selection recommendations

This study uses two public datasets, Movielens-100K and Amazon, for comparative analysis, and uses a combination of offline and simulated online testing to evaluate the model. This study uses the user's rating of the product (1-5 points) in the dataset as a feedback signal. Among them, all interactions with click behavior are counted as positive feedback, and the record with the highest score of 5 points is specifically defined as high-scoring feedback that represents the user's special love. The Movielens-100K dataset covers nearly a thousand users and more than a thousand products, with 100,000 interaction records. Among them, the proportion of high-score feedback is relatively high. The Amazon dataset is larger, with thousands of users and thousands of products, the total number of interactions exceeds 150,000, and the number of high-scoring feedback is relatively high. Although Movielens-100K and Amazon are static datasets that cannot fully capture the dynamics of live e-commerce, they can serve as appropriate benchmarks for evaluating the core capabilities of the model in collaborative filtering and sequential interest modeling. These features are the foundation of any recommendation system used in dynamic environments. To ensure transparency and reproducibility, this section details the evaluation setup employed in the experiments. Candidate Generation: For each user, the candidate set consists of one positive item and 2,000 randomly sampled negative items. Evaluation Metrics: All metrics are reported as averages over test users. The primary metrics include Accuracy@K: proportion of users for whom the actual item appears in the top-K recommendations (K=20). Recall@K: proportion of correctly recommended items relative to all positive items. F1@K: harmonic mean of Accuracy@K and Recall@K. NDCG@K: normalized discounted cumulative gain, accounting for ranking positions. Hit Rate@K: proportion of users with at least one hit in the top K list. The "recommendation accuracy" reported in Figure 7 refers specifically to top 1 hit rate (*i.e.*, Accuracy@1), measuring whether the model places the target item at the top of the candidate list. To assess the robustness of observed differences, paired bootstrap resampling (1,000 iterations) is applied to compute 95% confidence intervals for all metrics. Statistical significance between models is determined by whether the confidence interval of the metric difference excludes zero. This study compares and analyzes the recommendation accuracy of Discrete Sequential Set Rec-

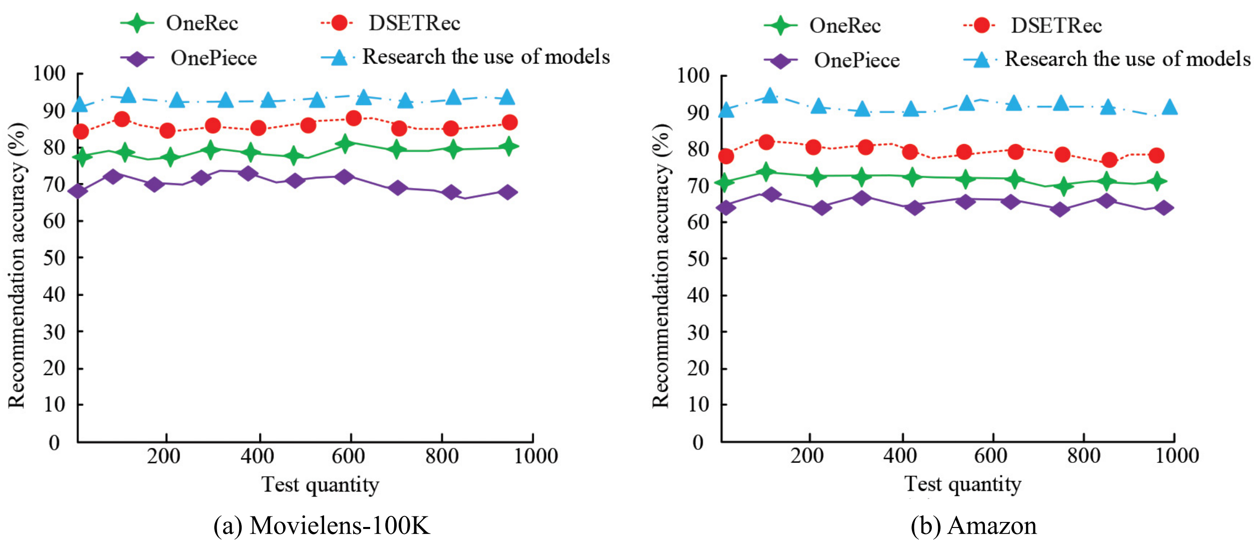
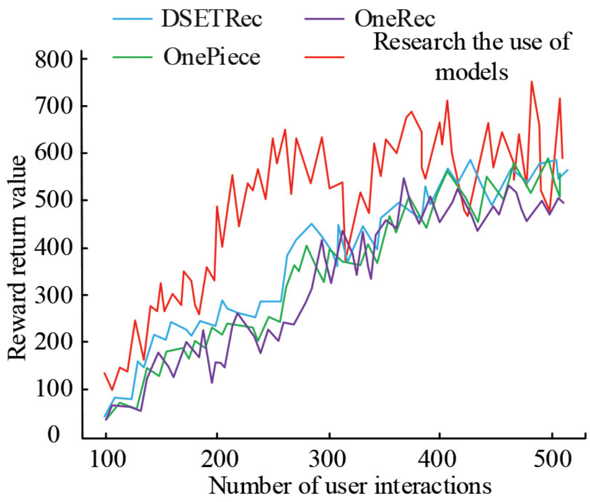


Figure 7. Comparison of accuracy of product recommendation on different datasets.

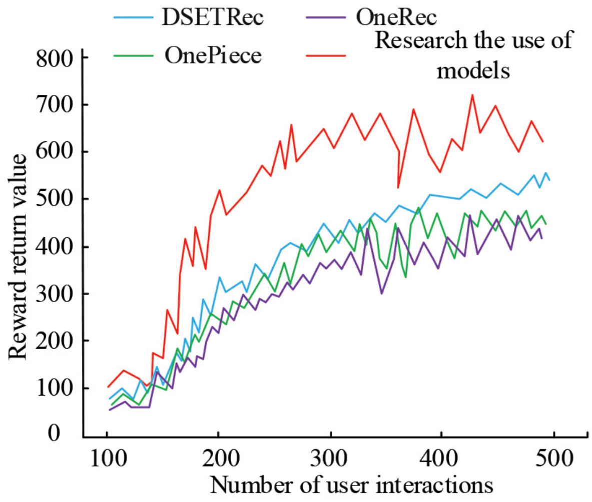
ommendation (DSETRec), OnePiece model, and One Recommendation (OneRec) from a set perspective. Among them, two datasets are analyzed, and the number of test products is 1,000. The comparative analysis is shown in Figure 7.

In Figure 7(a), in the dataset Movielens-100K test, the test accuracy of different models fluctuates up and down as the number of products increases. Among them, the test accuracy of the research model is the highest, reaching a maximum of 92.6%. OnePiece's recommendation accuracy is the lowest among several models. Its highest value is only 74.8%, which is 17.8% lower than the recommended model used in the study. In Figure 7(b), in the Amazon comparison of the dataset, the recommendation accuracy of the research model can reach up to 92.5%, while the recommendation accuracy of the OnePiece model is only 66.7%, which is 25.8% lower than the research model. This shows that among the different model recommendation tests, the recommendation model used in this study performs better. This is because the research model uses multi-agent algorithms and deep reinforcement learning, which improves the model's ability to extract and integrate user information. The study conducts reward regression analysis on different models, as shown in Figure 8. Among them, the higher the regression reward value directly indicates the better the overall performance of the agent in the experiment.

In Figure 8(a), in the dataset Movielens-100K test, the regression reward value of different models gradually increases with the increase of user interaction time. Among them, the regression reward value of the model used in this study is the highest, reaching a maximum of 736, while the regression reward value of the OneRec model is relatively low. Its highest value is only 512, which is 224 lower than the research model. In Figure 8(b), in the Amazon test of the dataset, the regression reward value of the model used in the research is significantly reduced. Among them, the highest value is only 711. The regression reward value of the OneRec model is also lower among several models. Among them, the highest value is only 416, which is about 295 lower than the research model. In summary, the regression reward value of the model used in the research on different model tests is higher, which shows that there is a better test effect in the overall model agent running interaction detection. This study compares and analyzes the running test results of different models, as shown in Table 1. In addition to the three metrics reported, complementary evaluation on NDCG@20 and Hit Rate@20 yields consistent patterns: the proposed model achieves NDCG@20 of 0.2156 on Movielens and 0.0934 on Amazon, and Hit Rate@20 of 0.1923 and 0.0765 respectively, outperforming all baselines.



(a) Movielens-100K



(b) Amazon

Figure 8. Comparison results of regression reward values for different models.

Table 1. Comparison of test results for different model operations.

Methods	Movielens-100K			Amazon		
	Accuracy@20	Recall@20	F1@20	Accuracy@20	Recall@20	F1@20
DSETRec	0.1852	0.1046	0.1195	0.0674	0.0663	0.0621
OnePiece	0.1804	0.1028	0.117	0.0642	0.0635	0.0608
OneRec	0.1826	0.1039	0.1183	0.0681	0.0672	0.0635
Research the use of models	0.2051	0.1185	0.1332	0.0803	0.0792	0.0748

In Table 1, on the Movielens dataset, different models show significant differences in Accuracy@20, Recall@20, and F1@20 indicators. Among them, the Accuracy@20 of the research model is the highest, reaching 0.2051. The performance of the OnePiece model is relatively low, with an Accuracy@20 of only 0.1804, which is about 0.0247 lower than the model used in the research. On the Amazon dataset, the overall indicators of each model decline compared with Movielens. The Accuracy@20 of the research model is 0.0803, and the Accuracy@20 of the DSETRec model on this dataset is 0.0674, which is about 0.0129 lower than the research model. In summary, under different datasets and evaluation indicators, the models used in the study all show better comprehensive recommendation performance. This shows that it has better stability and generalization ability in capturing user preferences and item associations.

4.2. Comparison of model ablation experiment and model interaction test

This study conducts online evaluation in a simulation environment, using warm start experimental settings for e-commerce datasets, and taking the user's first 15 clicks at the current moment as the initial state. Moreover, 2,000 non-interacted items are randomly selected as negative samples, which together with the positive samples form an alternative set. The recommendation system pushes one product at each moment, with a total of 20 steps in each round of interaction. Training and testing use 80% and 20% of user data, respectively. In terms of the model, a graph embedding network is used to map the heterogeneous graph into a 32-dimensional vector, the node neighborhood sampling size is 5, and the discount rate is set to 0.85. Research builds a simulated online recommendation service environment, with hardware configuration of Intel Xeon Gold 6248R (24 cores, 2.7 GHz) CPU and 128 GB of memory, software using Ubuntu 22.04, PyTorch 2.1.0, and TorchServe, and caching candidate sets and storage features using Redis and MySQL respectively. Using JMeter for load testing, the number of threads increases from 50 to 800, with 30 cycles per thread. After 200 rounds of preheating, response time, throughput, and resource utilization are collected. The construction of the candidate set is consistent with offline evaluation. Each request extracts 2000 non-interacting items and real target items to form a candidate set. The model outputs the item with the highest Q value to ensure response time QPS. The indicators such as concurrent support can be replicated. This study compares DSETRec, OnePiece, and OneRec algorithm models for user interaction, as shown in Figure 9. The shorter the user interaction time, the higher the product utilization rate of the entire model. It analyzes the actual application effect of the model through changes in user interaction time.

In Figure 9(a), in the dataset Movielens-100K test, the user interaction time of the research model is lower than that of other models. For several models, as the reward value increases, their user interaction time increases. Among them, the OnePiece model has more user interaction time among the compared models. The number of interactions reaches the maximum at 1.5 reward value, 38 user interaction times, which is an increase of 36 times compared to the research model. In Figure 9(b), in the Amazon test of the dataset, the user interaction time of the research model can reach up to 3 times, which is 26 times lower than the OnePiece model. In summary, in different model tests, the user interaction time of the model used in the study is less, and the overall product recommendation utilization rate is higher. This study compares the model testing accuracy after introducing different modules, as shown in Figure 10.

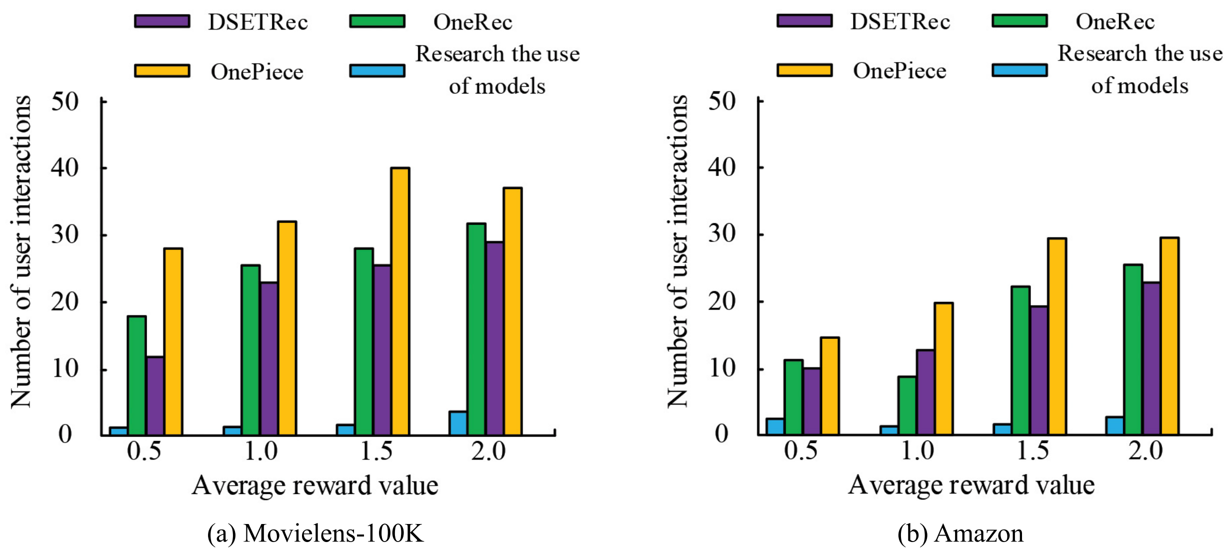


Figure 9. Comparison of user interaction times.

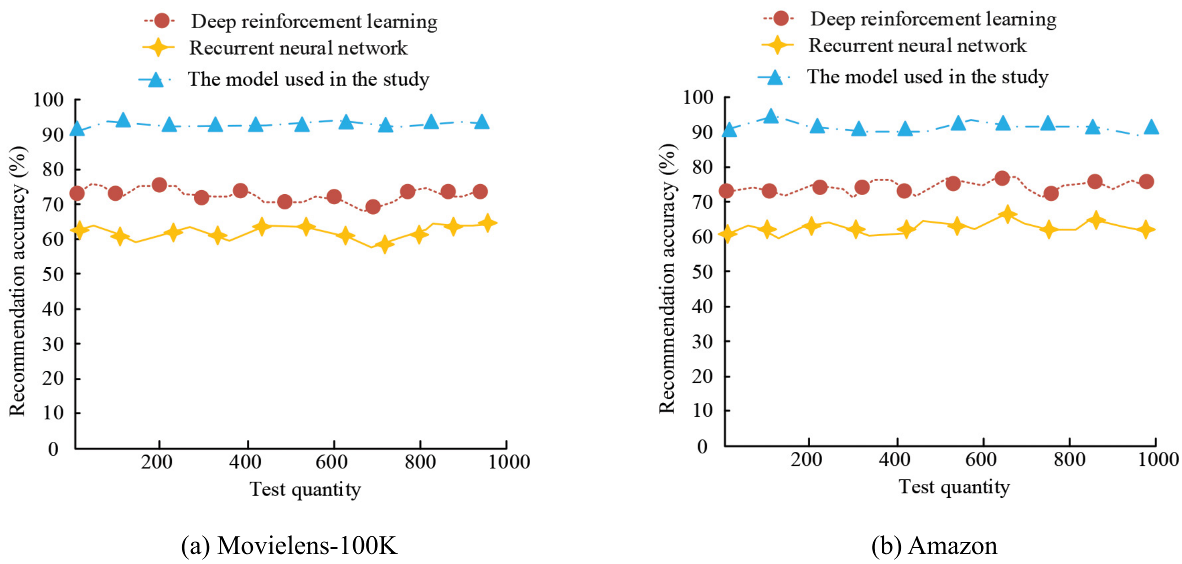


Figure 10. Test accuracy ablation experiment.

In Figure 10(a), the test accuracy of the model used in different ablation experiments has been significantly improved after introducing different network modules. Among them, the accuracy of the research model increases by 28.7% compared with the deep reinforcement learning model. In Figure 10(b), in different data tests, the performance of traditional models is also lower than that of research models. Among them, the accuracy of the research model increases by 29.5% compared with the deep reinforcement learning model. In summary, the operating accuracy of the model used in the study is higher than that of other models. This study analyzes the actual test results of the model as shown in Table 2.

Table 2. The actual operational performance of the model.

Metric category	**Specific metric**	**Model performance (measured/simulated)**
Response performance	Average response time (P95)	0.18 seconds
	Recommendation load delay (P99)	0.32 seconds
Throughput & concurrency	Query throughput (QPS)	520 QPS
	Concurrent users supported	680
Resource efficiency	CPU usage (normal)	50%
	Memory usage	2.5 GB
Recommendation quality	NDCG@10	0.872
	Click-through rate (CTR)	6.20%
System stability	Availability (SLA, 7-day)	99.70%
Diversity & fairness	Product category coverage	78.20%

In Table 2, the recommended model proposed in this study shows superior performance suitable for actual deployment. Among them, the average response time of the research model is 0.18 seconds, the query throughput is 520 QPS, and it can support 680 concurrent users. Meanwhile, the resource usage efficiency is high, with a CPU usage of 50% and a memory consumption of 2.5 GB. In terms of recommendation quality, the model NDCG@10 reaches 0.872, and the simulated click-through rate is 6.2%, reflecting good user interest matching capabilities. The system has availability of 99.7% and a product category coverage of 78.2%. It has achieved a good balance between response speed, recommendation accuracy, system stability and operating economy. It is suitable for high-concurrency live broadcast e-commerce scenario applications. To verify the statistical significance of the observed differences, paired bootstrap resampling (1,000 iterations) is conducted on all metrics. The 95% confidence intervals for the proposed model's advantage over the best baseline (OneRec) on Movielens Accuracy@20 are [0.0182, 0.0315], confirming statistical significance. Similar results are obtained for other metrics and datasets.

5. Conclusion

This study proposes a personalized product recommendation framework for live-streaming e-commerce that integrates an improved QMIX. The key innovations include: (1) a multi-user information fusion module leveraging GATs and RNNs to produce unified team-level state representations from distributed user observations; and (2) a dynamic hybrid network that replaces QMIX's static mixing structure with an adaptive parameter generation mechanism driven by real-time team states. On top of this fusion backbone, a SAC-based deep reinforcement learning module with dual GRU encoders captures both short-term interaction dynamics and long-term user preferences, enabling continuous and stable recommendation strategy optimization. Experiments on the Movielens-100K and Amazon benchmark datasets confirm that the proposed model consistently outperforms baseline methods, including OneRec, OnePiece, and DSETRec, across all primary evaluation metrics, achieving Accuracy@20 values of 0.2051 and 0.0803 on the two datasets respectively and reducing user interaction time by up to 26 steps compared to the strongest baseline.

However, this study still has several limitations that need to be pointed out. The static offline datasets used for evaluation (Movielens-100K and Amazon datasets) failed to capture the real-time interactivity, content update frequency, and user behavior volatility characteristics in real-time streaming environments. Therefore, the results should be considered as reference conclusions rather than direct predictions of actual production effects. In addition, the multi-component architecture of this model leads to high training costs, and its computational efficiency in large-scale online deployment environments has not yet been verified by empirical studies.

Future research will focus on two main directions: First, to conduct pilot deployments and online A/B testing of the framework on real live-streaming e-commerce platforms to verify its effectiveness in real-world scenarios; second, to reduce service latency through model compression and inference acceleration technologies, thereby expanding the framework's applicability in resource-constrained deployment scenarios.

Declaration of Competing Interests

The authors declare no conflict of interest.

Funding

This work was supported by the Special Project for Rural Revitalization of Gansu Provincial Department of Science and Technology: "Mechanism and Pathway of Digital Empowerment for Upgrading the 'Specialty Local Products' Industry Chain in Gansu Province" (Grant No. 25CXNM004); the Doctoral Research Foundation of Longdong University: "Research on the Pathway of Digital Economy Promoting Rural Industry Revitalization in Gansu Province under the Context of Common Prosperity" (Grant No. XYBYSK2207); the Soft Science Project of Gansu Provincial Department of Science and Technology: "Research on the Pathway of 'East Data West Computing' Project Promoting Rural Industry Revitalization in Gansu Province under the Context of Common Prosperity" (Grant No. 23JRZA497); and the Gansu Provincial Social Science Foundation Project: "Research on the Policy System for Cultivating and Strengthening Small and Medium-Sized Enterprises That Specialize in Novelty and Refinement in Gansu Province" (Grant No. 2024YB119).

Data Availability

Data used in this article are available upon request from the authors.

Reference

[1] D. N. Nguyen, V. H. Nguyen, T. Trinh, T. Ho, and H. S. Le, "A personalized product recommendation model in e-commerce based on retrieval strategy," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 2, p. 100303, 2024. http://dx.doi.org/10.1016/j.joitmc.2024.100303

[2] I. Karabila, N. Darraz, A. EL-Ansari, N. Alami and M. EL Mallahi, "BERT-enhanced sentiment analysis for personalized e-commerce recommendations", *Multimedia Tools and Applications*, vol. 83, no. 19, pp. 56463-56488, 2024. https://doi.org/10.1007/s11042-023-17689-5

[3] M. S. Bhuiyan, "The role of AI-Enhanced personalization in customer experiences", *Journal of Computer Science and Technology Studies*, vol. 6, no. 1, pp. 162–169, 2024. https://doi.org/10.32996/jcsts.2024.6.1.17

[4] C. P. Gupta, V. V. R. Kumar and A. Khurana, "Artificial intelligence application in e-commerce: Transforming customer service, personalization and marketing", in *Proceedings of the 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 10–16, 2024. https://doi.org/10.23919/INDIACom61295.2024.10498474

[5] W. Li, Y. Cai, M. H. Hanafiah and Z. Liao, "An empirical study on personalized product recommendation based on cross-border e-commerce customer data analysis", *Journal of Organizational and End User Computing (JOEUC)*, vol. 36, no. 1, pp. 1-16, 2024. https://doi.org/10.4018/JOEUC.335498

[6] F. Messaoudi and M. Loukili, "E-commerce personalized recommendations: a deep neural collaborative filtering approach", *Operations Research Forum*, vol. 5, no. 1, p. 5, 2024. https://doi.org/10.1007/s43069-023-00286-5

[7] W. Jin, H. Du, B. Zhao, X. Tian, B. Shi and G. Yang, "A comprehensive survey on multi-agent cooperative decision-making: Scenarios, approaches, challenges and perspectives", *arXiv preprint arXiv:2503.13415*, 2025. https://doi.org/10.48550/arXiv.2503.13415

[8] J. D. Bodapati, N. Veeranjaneyulu and N. K. Yenduri, "A comprehensive multi-modal approach for enhanced product recommendations based on customer habits", *Journal of The Institution of Engineers (India): Series B*, vol. 105, no. 6, pp. 1537–1545, 2024. https://doi.org/10.1007/s40031-024-01064-5

[9] N. M. U. Din, S. U. Sabha, T. Majeed, S. Rafiq and A. Assad, "Deep Reinforcement Learning in Medical Science: Methods, Applications, and Future Directions", in Artificial Intelligence in Human Health and Diseases, pp. 47–63, 2025. https://doi.org/10.1007/978-981-96-8176-1_3

[10] W. Wang, Y. Tang and K. Tian, "LeadRec: Towards Personalized Sequential Recommendation via Guided Diffusion", in *Proceedings of the International Conference on Intelligent Computing*, pp. 3–15, 2024. https://doi.org/10.1007/978-981-97-5618-6_1

[11] J. Chen, "Design and Implementation of a Personalized Recommendation System Based on Deep Learning Distributed Collaborative Filtering Algorithm on Social Media Platforms", in *Proceedings of the 2025 3rd International Conference on Integrated Circuits and Communication Systems (ICICACS)*, pp. 1–5, 2025. https://doi.org/10.1109/ICICACS65178.2025.10967882

[12] S. Kalra, A. Faiz, D. Aggarwal, *et al.*, "Optimizing CNNR-NNT model for effective product recommendation in e-commerce," in *Proceedings of the 2025 International Conference on AI-Driven STEM Education and Learning Technologies (AISTEMEDU)*, vol. 25, no. 10, pp. 1–7, 2025. http://dx.doi.org/10.1109/AISTEMEDU67077.2025.11403941

[13] Y. Yin, Y. Li, H. Yu *et al.*, "Contribution-Aware Personalized Federated Learning for Consumer Electronics Products Trustworthy Recommendation", *IEEE Transactions on Consumer Electronics*, vol. 22, no. 11, pp. 7241–7253, 2025. https://doi.org/10.1109/TCE.2025.3544884

[14] W. Chen, Z. Shen, Y. Pan, K. Tan and C. Wang, "Applying machine learning algorithm to optimize personalized education recommendation system", *Journal of Theory and Practice of Engineering Science*, vol. 4, no. 01, pp. 101–108, 2024. https://doi.org/10.53469/jtpes.2024.04(01).14

[15] S. Li, K. Liu and X. Chen, "A context-aware personalized recommendation framework integrating user clustering and BERT-based sentiment analysis", *Journal of Computer, Signal, and System Research*, vol. 2, no. 6, pp. 100-108, 2025. https://doi.org/10.71222/1cgq9333

[16] M. L. Goh, S. H. Tan, A. T. Shin, and, W. G. P. Hidayat, "Environmental Characteristics, Service Quality, and Brand Image as Drivers of Customer Loyalty in Malaysian Book Cafés", *Journal of Logistics, Informatics and Service Science*, vol. 13, no. 1, pp. 98–116, 2026. http://dx.doi.org/10.33168/JLISS.2026.0105

[17] M. Bismay, X. Dong and J. Caverlee, "Reasoningrec: Bridging personalized recommendations and human-interpretable explanations through llm reasoning", in *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8132–8148, 2025.
https://doi.org/10.18653/v1/2025.findings-naacl.454

[18] M. Li, Z. Zhang, M. Lu, X. Jia, R. Liu, X. Zhou and Y. Zhang, "Internet Financial Credit Risk Assessment with Sliding Window and Attention Mechanism LSTM Model", *Tehnicki vjesnik–Technical Gazette*, vol. 30, no. 1, pp. 1–7, 2023.
https://doi.org/10.17559/TV-20221110173532

[19] R. Pelánek, T. Effenberger and P. Jarušek, "Personalized recommendations for learning activities in online environments: a modular rule-based approach", *User Modeling and User-Adapted Interaction*, vol. 34, no. 4, pp. 1399–1430, 2024.
https://doi.org/10.1007/s11257-024-09396-z

[20] J. Chen, Z. Liu, X. Huang, C. Wu, Q. Liu, G. Jiang *et al.*, "When large language models meet personalization: Perspectives of challenges and opportunities", *World Wide Web*, vol. 27, no. 4, pp. 42–43, 2024.
https://doi.org/10.1007/s11280-024-01276-1

[21] P. Zhao, Y. Jin, X. Ren *et al.*, "A personalized cross-domain recommendation with federated meta learning", *Multimedia Tools and Applications*, vol. 83, no. 28, pp. 71435–71450, 2024.
https://doi.org/10.1007/s11042-024-18495-3

[22] Z. H. Wei, L. Yan and X. Yan, "Optimizing Production with Deep Reinforcement Learning", *Int. Journal of Simulation Modelling*, vol. 23, no. 4, pp. 692–703, 2024.
https://doi.org/10.2507/IJSIMM23-4-CO17

[23] F. Morton, T. T. Benavides and E. González-Treviño, "Taking customer-centricity to new heights: exploring the intersection of AI, hyper-personalization, and customer-centricity in organizations", in *Smart Engineering Management*, pp. 23–41, 2024.
https://doi.org/10.1007/978-3-031-52990-0_2

[24] X. Wang, "The Impact of Logistics Information Technology on Consumer Perceptions of Personalized Marketing in Omnichannel Supply Chains", *Journal of Logistics, Informatics and Service Science*, vol. 13, no. 1, pp. 274–291, 2026.
http://dx.doi.org/10.33168/JLISS.2026.0115

[25] J. Zhao, "Multi-Agent Reinforcement Learning with Graph Convolutional Networks for Collaborative Task Scheduling in Distributed Virtual Reality Systems," *CIT. Journal of Computing and Information Technology*, vol. 1, no. 4, pp. 249–266, 2026.
http://dx.doi.org/10.20532/cit.2025.1006202

[26] H. Shu, "Towards Robust Urban Spatial Recognition and Dynamic Optimization: A Multimodal Spatiotemporal Neural Network Approach," *CIT. Journal of Computing and Information Technology*, vol. 1, no. 4, pp. 231–248, 2026.
http://dx.doi.org/10.20532/cit.2025.1006157

Received: February 2026
Revised: March 2026
Accepted: April 2026

Contact addresses:

Zihan Yang
Woosuk University
School of Economics and Management
Longdong University
Qingyang 745000
China
e-mail: yangzh9901@163.com

Sibo Wang
Xi'an Physical Education University
School of Education
Longdong University
Qingyang 745000
China
e-mail: zggsqywsb@163.com

Bingjie Hou
Nottingham Trent University (UK)
School of Economics and Management
Longdong University
Qingyang 745000
China
e-mail: houbj@ldxy.edu.cn

Jianming He
La Trobe University (Australia)
School of Economics and Management
Longdong University
Qingyang 745000
China
e-mail: hejm@ldxy.edu.cn

ZIHAN YANG received a Ph.D. in Business Administration from Woosuk University, South Korea in August 2020. He is currently an Associate Professor at the School of Economics and Management, Longdong University, China. His research interests include digital economy.

WANG SIBO received a master's degree in Physical Education from Xi'an Physical Education University in July 2018. He is currently a Lecturer at the School of Education, Longdong University, China. His research interests include sports industry.

BINGJIE HOU obtained a master's degree in human resource management from the Nottingham Trent University, United Kingdom, in November 2013. She is currently an Associate Professor at the School of Economics and Management, Longdong University, China. Her research interests include digital economy.

JIANMING HE received a master's degree in international business from the La Trobe University, Australia, in October 2012. He is currently an Associate Professor at the School of Economics and Management, Longdong University, China. His research interests include digital economy.