Optimizing A Reinforcement Learning Recommendation Algorithm For Personalized English Vocabulary Learning

Xin GUO[1] and Si CHEN[2*]

[1] Qinhuangdao Open University, Qinhuangdao City, Hebei Province, 066000, China
[2] Xingtai Open University, Xingtai City, Hebei Province, 054000, China
[*] Corresponding author. E-mail: chensi202571@163.com

Received: Sep. 21, 2025; Accepted: May. 16, 2026

To address the challenges of personalized English vocabulary learning, this paper proposes a reinforcement learning recommendation algorithm optimization method that aims to balance dynamic user feature modeling with long-term policy stability. Traditional methods often face the problem of recommendations favoring short-term goals while neglecting long term learning effects. This is particularly true for English vocabulary learning, as the influence of the Ebbinghaus forgetting curve further illustrates this problem. To this end, this paper designs a dual-module collaborative framework that combines a dynamic feature encoder with a multi-time-scale policy network to optimize learning policies. User modeling uses a Transformer and GRU (Gate Recurrent Unit) architecture to construct long-term states, and the policy is optimized using the Deep Deterministic Policy Gradient (DDPG) framework. In terms of reward mechanism design, a multi-time-scale reward function is introduced, using a time decay factor to adjust the ratio of immediate and delayed rewards, balancing short-term memory and long-term knowledge accumulation. Experimental results demonstrate that the optimization method performs exceptionally well in the following areas: Top-10 accuracy reaches 88%; 7-day memory retention reaches 85% on the first day and remains stable; and in cold-start scenarios, the average first-time interaction coverage reaches 60%. Furthermore, the policy stability evaluation is only 0.02, validating the algorithm's efficiency and stability. This paper not only demonstrates algorithmic innovation but also has significant application value in the engineering implementation of personalized learning recommendation systems.

Keywords: reinforcement learning; personalized recommendation; English learning; time- scaled reward; deep deterministic policy gradient

http://dx.doi.org/10.6180/jase.202610_33.004

1. Introduction

With the rapid development of artificial intelligence and big data technologies, personalized recommendation systems have been widely used in education, particularly in language learning [1–3]. Optimizing personalized vocabulary recommendations is of great significance. Existing reinforcement learning recommendation algorithms struggle to balance dynamic user feature modeling with long-term learning strategy stability. This often results in recommendation systems failing to balance short-term learning preferences with long-term learning goals, which in turn impacts learning effectiveness [4, 5]. Therefore, achieving an efficient balance between short-term and long-term goals in personalized English vocabulary learning has become a key research issue.

This paper proposes a dual-module collaborative framework based on multi-time-scale reward functions, which addresses the short-term overfitting and long-term strat-

egy instability problems of reinforcement learning recommendation algorithms in personalized English vocabulary learning through three levels: cognitive-driven, domain adaptation, and engineering implementation. Dynamic feature encoders utilize self-attention to fuse multimodal features, generate low-dimensional short-term states, and accurately capture real-time behaviors and individual demands. The multi-time-scale policy network combines the Actor-Critic structure with the time decay factor to balance immediate feedback and long-term goals. In engineering, priority experience playback is adopted to accelerate convergence and enhance cold start adaptability and memory retention.

Experimental results demonstrate that this method demonstrates excellent accuracy, cognitive effectiveness, and adaptability. Specifically, the long-term memory retention rate reached 85% on the first day and remained stable thereafter. In cold-start scenarios, the average first-interaction coverage reached 60%, significantly outperforming baseline models such as traditional collaborative filtering and Q-learning. The policy stability evaluation was 0.02 , validating the effectiveness and engineering feasibility of this approach in personalized recommendation systems.

The core challenge of personalized English vocabulary learning recommendation systems lies in striking a balance between dynamic user state modeling, reinforcement learning strategy optimization, and long-term learning goals. Unlike general recommendation systems, English vocabulary learning is significantly language-specific: vocabulary memorization is subject to the Ebbinghaus forgetting curve, learning feedback is delayed, and knowledge points are semantically related. These characteristics make traditional recommendation methods less effective in vocabulary learning scenarios. While current research has made some progress in user modeling, strategy optimization, and goal balancing, three key flaws remain, urgently requiring systematic improvement.

The first major flaw is the lack of ability to decouple features across multiple timescales in dynamic user modeling. During English vocabulary learning, the user's state encompasses both immediate vocabulary mastery and the long-term trajectory of memory evolution. However, existing methods generally adopt a single time scale modeling, which makes it impossible to effectively capture the multi-stage memory characteristics in vocabulary learning. For example, the collaborative learning recommendation system proposed by Lahiassi J can dynamically adjust the group configuration, but it is difficult to model the changing trend of individual vocabulary memory strength in vocabulary learning [6]. Fariani R I reviewed the application of personalized learning in higher education, focusing on the analysis of learner diversity, personalized learning, development methods, and implementation impact, but did not involve an in-depth discussion of the dynamic modeling of user learning status [7]. Luo J proposed a dynamic multi-objective recommendation method based on discrete behavior soft actor-critic, which uses gated recurrent units to extract user features and design a composite reward function. However, it fails to explicitly model long-term memory states, which makes its performance in complex learning tasks relatively limited [8]. Although the bidirectional attention mechanism studied by Cheng *S* can integrate spatiotemporal preferences, its model is still limited to short-term interaction sequence modeling and ignores the long-term evolution of vocabulary memory [9]. The recommendation method based on dynamic collaborative filtering proposed by Wang H improves the personalization and speed of recommendations but is still limited to modeling a single time scale [10]. The above methods have significant shortcomings when modeling vocabulary memory curves and cannot meet the dynamic closedloop requirements of "memorization-forgetting-review" in English learning.

A second major flaw is that the reinforcement learning framework design ignores the delayed feedback characteristics of vocabulary learning. In English vocabulary learning, user feedback is often delayed, and reward signals are unstable. Existing research has mostly adopted reinforcement learning frameworks based on Q-learning or actor-criticism, but their reward function designs generally favor immediate feedback, ignoring the delayed consolidation effect of vocabulary memory. Although the constrained Markov decision process framework proposed by Shi X can optimize recommendation fairness, it is difficult to handle the delayed rewards caused by the memory curve in vocabulary learning [11]. The efficient elearning recommendation system based on the hybrid optimization algorithm proposed by Vedavathi N analyzes learner behavior and preferences to improve learning efficiency and meet learner needs [12]. The deep reinforcement learning recommendation method proposed by Li Z improves accuracy, but its strategy update mechanism does not take into account the vocabulary review cycle, and the recommendation strategy tends to be unstable in long-term learning [13]. In addition, the DDPG framework, due to its deterministic policy gradient update characteristics, has demonstrated rapid convergence in continuous action space control [14], but has not yet been effectively introduced into vocabulary learning scenarios to address the integration of multi-granularity reward signals.

The third major drawback is the lack of explicit modeling of long-term learning goals and memory curves. Current research mainly uses meta-learning or multi-agent collaborative strategies to model long-term learning goals, but these methods have obvious limitations in vocabulary learning. Mu M's empirical research shows that learning recommendation strategies that do not consider the Ebbinghaus forgetting curve may lead to inappropriate review timing and reduce memory retention [15]. Although the new cross-domain meta-learner framework proposed by Guan R improves the recommendation accuracy in cold start scenarios, its strategy update mechanism does not dynamically adjust the vocabulary memory curve, resulting in a disconnect between the recommended content and the user's forgetting rhythm [16]. In addition, although the multi-agent collaborative method can handle multi-time-scale goals, it has high computational overhead and deployment cost, making it difficult to meet the engineering feasibility requirements of actual education systems.

In summary, existing research faces three core problems in personalized English vocabulary learning recommendation systems: (1) Dynamic user modeling lacks a multi-timescale feature decoupling mechanism, making it difficult to characterize the evolution trajectory of vocabulary memory; (2) Reinforcement learning strategy design ignores the delayed feedback feature, resulting in recommendation strategies biased towards short-term goals; (3) The lack of explicit modeling of the memory curve makes it difficult to guarantee long-term learning results. These problems together constitute the "methodological blind spot" of current research, and there is an urgent need for systematic optimization from three aspects: model structure, reward mechanism, and engineering implementation. Meng introduced AVAR-RL, which employs an adaptive reward shaping mechanism based on learner performance trajectories, yet it does not incorporate long-term memory dynamics via the Ebbinghaus curve into the policy network and uses a single-scale GRU for state encoding, failing to decouple short-term interaction signals from long-term forgetting patterns [17]. In contrast, our dual module framework explicitly separates temporal abstraction layers through Transformer-based short-term attention and GRU-based long-term memory fusion, while embedding the forgetting curve directly into a multi-time-scale DDPG reward structure, thereby achieving more stable and cognitively grounded personalization.

This paper not only achieves innovation in model structure and reward mechanism but also provides a feasible solution at the engineering deployment level, providing solid theoretical and technical support for the practical application of a personalized English vocabulary learning recommendation system.

2. Methods

2.1. Cognition-Driven System Architecture

This paper enhances the effect of personalized English vocabulary learning by integrating dynamic feature coding and multi-time-scale strategy networks, taking into account both shortterm learning needs and long-term goals.

This paper designs a dynamic feature encoder based on the Transformer architecture and GRU to capture the key features of user behavior [18]. The multi-head self-attention mechanism of Transformer enhances the modeling ability of time series and improves the system's sensitivity to complex learning dynamics. GRU alleviates the problem of vanishing gradients and ensures the stable transfer of long-term features. The combination of the two effectively and accurately represents users' short-term needs, providing strong support for the recommendation system.

This paper designs a multi-time-scale policy network based on the DDPG architecture and combines the Actor-Critic dual network to achieve the collaborative optimization of short-term behavior and long-term goals [19]. The short-term Actor network selects recommended words, while the long-term Critic network assesses the long-term value of actions and provides feedback for optimization. This enables the system to enhance immediate effects while promoting knowledge accumulation, effectively avoiding short-term overfitting, and optimizing the long-term learning path under complex tasks.

Fig. 1 shows the overall framework of the system, including the structure and data flow of the dynamic feature encoder and the multi-time-scale policy network. Through this design, the recommendation strategy can be flexibly adjusted to adapt to the dynamic needs of users and optimize the long-term learning goals.

2.2. Adaptive State Representation Method

This paper proposes an adaptive state representation method based on multimodal feature fusion. By extracting features such as learning duration, forgetting curve, and vocabulary difficulty, low-dimensional embedding is carried out, and the feature weights are dynamically adjusted by using the self-attention mechanism to accurately depict the user's dynamic learning state and improve the recommendation accuracy.

During the feature extraction process, the user's learning duration, forgetting curve, vocabulary difficulty, learning frequency, and accuracy are jointly considered as input feature vectors $Xt = [T_t, E_t, D_t, F_t, A_t]$, where F_t denotes

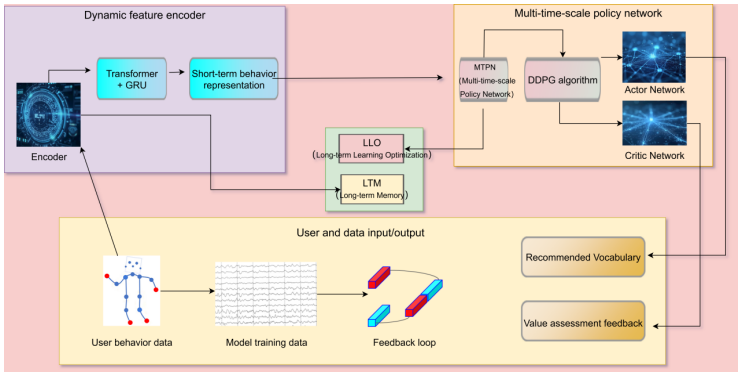

Fig. 1. Overall system framework

normalized learning frequency (number of prior exposures to the current word in the past 7 days) and A_t denotes standardized recall accuracy (percentage between 0 and 100%). All five dimensions follow the encoding rules in Table 1 and are concatenated before embedding. The vocabulary difficulty D_t is not computed by the model but is pre-assigned based on the Common European Framework of Reference for Languages (CEFR). Each word in the dataset is labeled with one of six discrete levels: A1, A2, B1, B2, C1, or C2, using authoritative lexical resources such as the English Vocabulary Profile. This categorical label is encoded as a 6-dimensional one-hot vector, consistent with the "Categorical+One-hot Encoding" rule specified in Table 1. No continuous or adaptive difficulty metric is generated during training; the model treats difficulty solely as a static, linguistically grounded category. To map these high-dimensional feature vectors into a lowdimensional space, this paper employs an embedding layer mapping. Let the embedding matrix be W . The resulting state representation is:

$$h_t = W \cdot X_t \tag{1}$$

In formula (1), h_t is the low-dimensional state representation vector, $W \in R^{d \times n}$ is the embedding matrix, d is the dimension of the embedding space, and n is the dimension of the feature vector.

Next, these features are weighted and fused through the self-attention mechanism [20]. For each time t , by calculating the weight α_t of each feature, the importance of different features in the state representation can be adaptively adjusted. Let the input feature be $X_t = [x_{1t}, x_{2t}, \ldots, x_{nt}]$, and the weight calculation formula in the self-attention mechanism is:

$$\alpha_{ij} = \frac{\exp\left(q_i \cdot k_j^T\right)}{\sum_{j=1}^{n} \exp\left(q_i \cdot k_j^T\right)} \tag{2}$$

In formula (2), q_i and k_j are the query vector and key vector, respectively, which represent the relationship between the input features. Then, based on these weights, the weighted sum of the features is the current state representation h_t :

$$h_t = \sum_{j=1}^{n} \alpha_{tj} \cdot x_{jt} \tag{3}$$

This weighted sum constitutes the final adaptive state representation h_t, which is directly fed into the actor and critic networks without any additional decoding step. The entire pipeline-from raw features through embedding, self-attention weighting, to state output-is end-to-end differentiable, ensuring that temporal dynamics and feature importance are jointly optimized during policy updates.

Fig. 2 shows the structure and process of the state representation model, which can further help people understand the feature extraction and fusion process.

Fig. 2 illustrates the process of gradually converting a user's multimodal features into a state vector. Through modules such as feature embedding, self-attention, and dynamic feature encoding, the user's multimodal information is extracted and fused. Ultimately, a vector representing the user's state is output.

The final state representation (h_t) is fed into the multi-time-scale policy network to further optimize the recommendation strategy using a reinforcement learning algorithm.

Although instantaneous feature values may coincide across users, personalization is preserved through two mechanisms: (i) the GRU in the dynamic feature encoder maintains a user-specific hidden state that evolves with each interaction, capturing individual learning trajectories; (ii) the self-attention weights in equation (2) are computed over each user's own historical sequence, making feature fusion context-dependent. Consequently, even with identical X_t, the resulting state representation differs due to

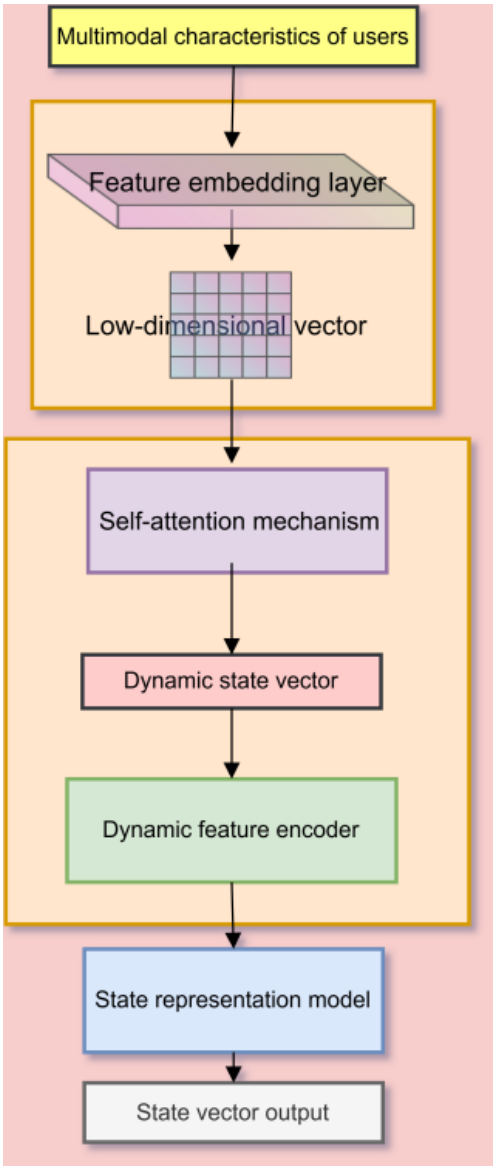

Fig. 2. State representation model structure

divergent sequential histories.

Table 1 shows the encoding methods of user characteristics in the personalized English vocabulary learning system: the learning duration and frequency are normalized, the forgetting curve is modeled through time series, the vocabulary difficulty is encoded in one-hot, and the accuracy rate is expressed as a standardized percentage. These multimodal coding methods effectively reduce user differences and enhance the accuracy of recommendations.

This approach allows the system to adjust the weights of different features in real time, effectively addressing temporal changes in user behavior. This allows the recommendation model to establish a more flexible and accurate connection between short-term user behavior and long-term learning goals.

2.3. Domain-Adapted Reward Function

To balance short-term rewards with long-term goals, this study employs a multi-time-scale reward function [21, 22], introduces a time decay factor to adjust the proportion of immediate and delayed rewards, and combines the Ebbinghaus forgetting curve to take into account both short-term feedback and long-term learning effects.

In this model, the short-term reward r_t reflects the immediate learning performance at time step t, such as correct vocabulary recall or task completion degree, and represents the learner's feedback on the current task. However, relying solely on short-term rewards can easily lead to the neglect of long-term memory accumulation. To this end, the long-term reward R_{t+H} is introduced to measure the return of the next H step and associate memory retention with knowledge accumulation. Combining the Ebbinghaus forgetting curve, long-term rewards effectively model memory decline and enhance the ability to evaluate long-term learning outcomes.

The comprehensive reward (R_t) is the weighted sum of the short-term reward (r_t) and the long-term reward, and its expression is:

$$R_t = r_t + \lambda \cdot \gamma^H \cdot R_{t+H} \tag{4}$$

In Formula (4), λ represents the balance factor between short-term and long-term rewards, $\gamma \in [0, 1]$ is the discount factor that adjusts the weight of future rewards, and H is the reward time span, indicating the return interval from t to $t + H$.

This paper designs a dynamic adjustment mechanism. As the user interaction history increases, the balance factor λ is gradually increased, and the discount factor γ is adjusted. In the initial stage, short-term feedback is emphasized, and in the later stage, the influence of longterm goals is enhanced. Combining the time decay factor (γ^H), the weight of the long- term reward is weakened to avoid short-sighted decision-making. This reward function effectively coordinates short-term behaviors with long-term memory goals, ADAPTS to the needs of different learning stages, and enhances the balance between knowledge accumulation and memory retention in personalized vocabulary learning.

2.4. Policy Optimization Process

This study adopts the DDPG algorithm combined with Prioritized Experience Replay (PER) for strategy optimization to enhance the long-term effect [23]. DDPG is suitable for

Table 1. State feature encoding rules

Feature Name	Feature Type	Encoding Method	Remarks
Learning Duration	Numeric	Normalization	Range: 0-1
Forgetting Curve	Curve	Time Series Modeling	Based on time decay
Vocabulary Difficulty	Categorical	One-hot Encoding	Classification of vocabulary
Learning Frequency	Numeric	Normalization	Frequency statistics
Accuracy	Percentage	Standardization	Between 0 and 100%

continuous action spaces and can precisely update long-term strategies based on user behavior. Compared with traditional methods, it is more applicable to high-dimensional and complex tasks, enhancing system performance and stability.

DDPG is based on the actor-critic structure [24, 25]. The actor generates policies, the Critic evaluates the quality of actions, and the state s_t and action a are mapped through a deep neural network. The update rules are as follows:

$$\theta_a^{t+1} = \theta_a^t + \alpha_a \nabla_{\theta_a} J_a \tag{5}$$

$$\theta_c^{t+1} = \theta_c^t + \alpha_c \nabla_{\theta_c} J_c \tag{6}$$

In formulas (5) and (6), θ_a and θ_c are the parameters of the Actor and Critic networks, respectively, α_a and α_c are the learning rates, and J_a and J_c are the corresponding loss functions. This structure achieves balanced optimization of short-term and long-term goals, enhances the stability of strategies, and ADAPTS to complex dynamic learning tasks.

To improve the stability of the strategy and the training efficiency, this paper introduces the priority experience replay mechanism [26–28]. According to the TD Error of Experience, samples with large errors and high influence are preferentially replayed to enhance the learning efficiency. By prioritizing experiences with larger errors for training, the system can accelerate the strategy update process and improve the long-term strategy stability. The TD error calculation formula is:

$$\delta_t = r_t + \gamma Q'\left(s_{t+1}, a_{t+1}; \theta_c\right) - Q\left(s_t, a_t; \theta_c\right) \tag{7}$$

In formula (7), $Q(s_t, a_t; \theta_c)$ is the state-action value function of the critic network, $Q'(s_{t+1}, a_{t+1}; \theta_c)$ is the value function of the target critic network, r_t is the immediate reward, and γ is the discount factor. By prioritizing experiences based on the TD error, high-priority experiences in the experience replay pool can be sampled more frequently, ensuring that the algorithm pays more attention to experiences that have a greater impact on long-term returns during training, thereby accelerating policy convergence and effectively improving the stability of the long-term policy.

The core of policy optimization lies in updating the parameters of the actor and critic networks through multiple iterations. The algorithm iteratively updates the strategy based on environmental feedback to maximize cumulative rewards. After introducing priority experience playback, key experiences that have a significant impact on long-term goals are sampled with emphasis, accelerating convergence and enhancing the stability of strategies and learning efficiency.

2.5. Experimental Design

This study utilizes the Kaggle public dataset "Educational Attainment in English Towns," containing anonymized logs from 12,847 learners interacting with 5,210 vocabulary items over 90 days. Each log includes user ID, word ID, timestamp, binary correctness, session duration, and confidence score. Preprocessing includes: (i) filtering sessions < 10 seconds; (ii) grouping interactions by user and sorting chronologically; (iii) computing learning frequency as 7-day exposure count; (iv) calculating accuracy as exponential moving average of past correct responses (decay=0.9); (v) deriving forgetting curve parameter via based on inter-exposure interval. All features are then encoded per Table 1 and aligned with model inputs. To ensure a rigorous comparison, we include both classical and recent baselines: collaborative filtering (CF), Q-learning, DQN, SAC, and AVAR-RL. All models use identical feature inputs, train on the same data splits, and undergo hyperparameter tuning via grid search on the validation set. All of them adopted standard implementation and optimized hyperparameters to ensure fair comparison.

The experimental evaluation indicators include accuracy rate (Top-N), coverage rate, stability, policy convergence, and computational efficiency. Accuracy measures the matching degree between the recommended results and the real records, while coverage reflects the ability of the recommended vocabulary to cover user needs. Stability is quantified by the KL divergence (Kullback-Leibler Divergence) to quantify the convergence of the strategy during training. The KL divergence formula is:

$$\mathrm{KL}\left(\pi_t || \pi_{t+1}\right) = \sum_a \pi_t(a \mid s) \log\left(\frac{\pi_t(a \mid s)}{\pi_{t+1}(a \mid s)}\right) \tag{8}$$

Formula (8) measures the difference in policy distribution between two consecutive iter ations; smaller values indicate more stable policy convergence. Computational efficiency is measured by recording the average time spent by different algorithms in each training iteration to verify their engineering feasibility in practical applications. All model hyperparameters are tuned through grid search and cross-validation. For this algorithm, the adjustment of the balance factor λ and the discount factor γ in the reward function, are optimized as key hyperparameters to improve recommendation effectiveness and long-term adaptability.

Table 2 shows the dataset partitioning and associated parameter settings. In this experiment, the dataset was divided into training, validation, and test sets, with ratios of 70%, 15%, and 15%, respectively. Each dataset was processed using the same hyperparameter settings, including learning rate, batch size, number of network layers, and number of hidden units. Regarding feature processing, all datasets were normalized and one-hot encoded to ensure that different features had similar scales and could effectively participate in model training. These settings ensured uniformity in the experimental process and facilitated fair evaluation of different models.

3. Results and discussion

The recommendation accuracy is evaluated by the Top- N accuracy rate ($N = 10, 20$) [29, 30]. The system generates the recommendation list and compares it with the user's actual learning record to calculate the proportion of matching words. A match is regarded as a correct recommendation.

The Top-10 and Top-20 accuracies are defined as:

$$\text{Top-}N \text{ Accuracy} = \frac{\text{Number of Correct Recommendations}}{N} \tag{9}$$

In Formula (9), N is taken as 10 and 20 to measure the accuracy of the first N recommendations. The performance of the proposed algorithm in personalized recommendation is evaluated by comparing it with the baseline model. In the experiment, the algorithm dynamically recommends based on the user's progress and features, enhancing the accuracy of recommendations. This indicates that the system is more in line with personalized needs, reduces redundancy and bias, improves learning efficiency, and quantifies the application effect of the model in actual scenarios.

Fig. 3 shows a comparison of different algorithms in terms of Top-10 and Top-20 accuracy. The horizontal axis represents the different algorithms, and the vertical axis represents accuracy, indicating the percentage of the recommendation system's top N recommended items that correctly predict the user's learning vocabulary. As the algorithm's performance improves, the Top-10 and Top-20 accuracy rates gradually increase. The proposed optimization algorithm, DDPG, achieved a Top-10 accuracy of 0.88 and a Top-20 accuracy of 0.92 , significantly exceeding other baseline algorithms. Traditional collaborative filtering algorithms, unable to effectively capture user dynamics, performed poorly on these Top-10 and Top-20 scores. Deep learning-based algorithms such as Q-learning, DQN, and SAC, while achieving improved performance, still failed to surpass the performance of the DDPG algorithm. This demonstrates that the proposed dual-module collaborative framework, through its combination of a dynamic feature encoder and a multi-time-scale policy network, can better understand and adapt to users' learning needs, particularly in balancing short-term and longterm goals, ultimately improving the accuracy of the recommendation system.

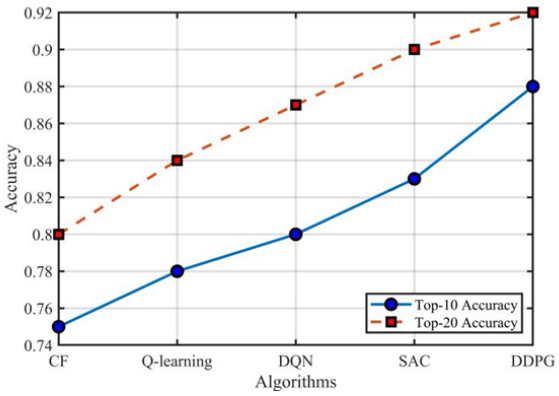

Fig. 3. Top-N accuracy comparison line chart

3.1. Long-Term Goal Adaptability Evaluation

In the long-term goal adaptability evaluation, the core objective is to measure the algorithm's ability to support long-term memory and user retention. To this end, the paper measured the accuracy of users' repetition of recommended words within seven days of learning. Based on the user's actual learning history over the past seven days, the system calculates the user's repetition rate of system-recommended vocabulary and compares this rate with the algorithm's recommendations.

The algorithm's performance is quantified by the seven-day retention rate, which reflects the user's ability to retain vocabulary over a certain time. The calculation formula is:

Table 2. Dataset division and parameter settings

Data Split	Percentage	Parameters	Feature Processing
Training Set	70%	Learning Rate: 0.001, Batch Size: 64, Network Layers: 3, Hidden Units: 128	Standardization, One-hot Encoding
Validation Set	15%	Learning Rate: 0.001, Batch Size: 64, Network Layers: 3, Hidden Units: 128	Standardization, One-hot Encoding
Test Set	15%	Learning Rate: 0.001, Batch Size: 64, Network Layers: 3, Hidden Units: 128	Standardization, One-hot Encoding

$$\text{Memory Retention Rate} = \frac{\text{Number of Correctly Recalled Words}}{\text{Total Number of Recommended Words}} \tag{10}$$

Formula (10) is used to evaluate whether the recommendation system can maintain the user's long-term memory of recommended words through long-term strategy optimization. The DDPG algorithm introduces a multi-time-scale reward function, combining immediate rewards and delayed rewards, to balance the relationship between short-term learning effects and longterm memory retention, thereby effectively promoting the adaptation of long-term goals.

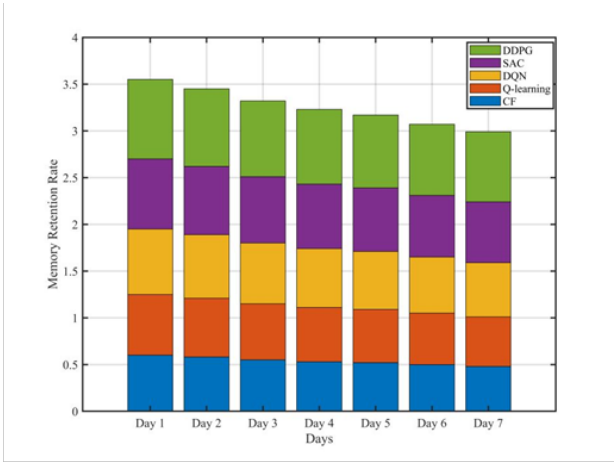

Fig. 4. Long-term memory retention comparison bar chart

Fig. 4 shows a comparison of the daily memory retention rates of different algorithms over seven days. DDPG's memory retention rate was 0.85 on the first day. While this rate gradually decreased over time, it remained high, demonstrating its strength in long-term memory retention. DDPG performed well in each day's memory retention rate, with particularly high retention rates in the first few days, demonstrating its effectiveness in maintaining long-term memory of learning content. Traditional algorithms (such as CF and Q- learning) have low retention rates and weak long-term adaptability. But the dual-module collaborative framework proposed in this paper effectively balances short-term and long-term rewards through multi-time-scale reward functions, improves memory retention ability and cold start adaptability, and optimizes the long-term effect of the recommendation system.

3.2. Cold Start Scenario Effectiveness Evaluation

To evaluate the effectiveness under cold start, this paper uses the recommendation coverage rate after the first five interactions of new users as an indicator to test the algorithm's adaptability to new users and the effect of vocabulary recommendation in the absence of historical data.

The specific method is as follows: First, several new users are randomly selected from the dataset and, without relying on any historical data, the first five recommendation interactions are performed. After each recommendation, the percentage of recommended vocabulary items that the user has actually learned is calculated as a measure of recommendation coverage.

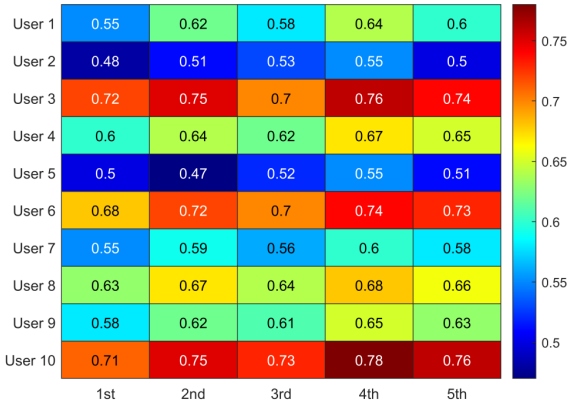

Fig. 5. Heatmap of recommendation coverage for new users

Fig. 5 shows the recommendation coverage for new users. The horizontal axis represents the number of interactions between the recommendation system and the user, and the vertical axis represents the number of different users. The average recommendation coverage for the first interaction with 10 users reached 0.6 . As the num-

ber of interactions increases, the recommendation coverage generally increases. During the initial cold start phase, a recommendation system relies on limited user information, resulting in relatively low recommendation coverage. As users interact with the system more frequently, the system gradually accumulates more user behavior data, enabling it to better predict user needs and provide personalized recommendations. Fig. 5 shows that user 3 and user 10 have higher coverage, indicating that the system has learned their preferences early on. For other users, especially users 2 and 5, the system requires more interaction data to adapt to their learning needs, demonstrating the gradual adaptation of the recommender system when dealing with the cold-start problem. This mechanism is closely related to the dynamic feature encoder and multi-time-scale policy network described in this paper. Through time decay factors and state modeling, it effectively balances immediate rewards and long-term goals, thereby gradually optimizing recommendation accuracy during the cold-start phase.

3.3. Policy Stability Evaluation

To evaluate the algorithm's policy stability, this paper uses KL divergence to quantify the distribution differences between recommended policies in adjacent iterations [31, 32]. KL divergence is a metric that measures the similarity between two probability distributions; smaller values indicate greater stability of the recommended policy. After each policy update, the KL divergence between the current policy and the previous policy is calculated, and the trend of change over multiple iterations is analyzed.

In the experiment, the model was first trained through multiple iterations, and the policy distribution after each update was recorded. Then, for each pair of adjacent iterations, the KL divergence between the policy distributions is calculated. By observing the changing trends of the KL divergence, it can analyze whether the algorithm is converging during training, as well as the speed and stability of convergence. If the KL divergence approaches 0 , it indicates that the policy has stabilized and the system has converged. If the KL divergence fluctuates significantly, it indicates that the policy is still being adjusted and the system has not yet reached convergence.

Fig. 6 shows the KL divergence changes of different algorithms over multiple training cycles, reflecting the oscillation amplitude of each algorithm's strategy. The horizontal axis represents the number of training iterations, totaling 50 cycles, while the vertical axis represents the KL divergence value, which measures the degree of strategy change for each algorithm. The KL divergence of DDPG is low (about 0.02 after training), the strategy update is smooth,

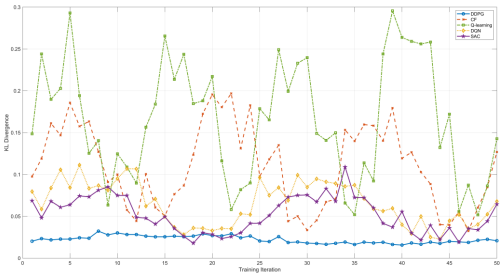

Fig. 6. Strategy oscillation amplitude time series curve

and the stability is high. However, Q-learning and CF have large fluctuations and slow convergence. Due to the lack of an effective mechanism to balance exploration and utilization, the early strategies frequently oscillate. DDPG relies on deep reinforcement learning and deterministic policy gradients to achieve smooth optimization of continuous action spaces, converging to stable strategies earlier and enhancing training efficiency and stability.

3.4. Computational Performance Evaluation

To assess the engineering feasibility, this paper records the average time consumption of each algorithm in a single round of training iterations. Through multiple experiments, the average training time is compared to measure the occupation of computing resources. The results are shown in Table 3 :

Table 3 shows the computational resources consumed and the time efficiency of different recommendation algorithms during training. It can be seen from Table 3 that the computational overhead of DDPG and SAC is relatively large. The training times are 1.85s and 1.6s, respectively, and the convergence times reach 35 min and 45 min . The CF is the lowest, with only 0.72 seconds of training, 512 MB of memory, and 15 minutes of convergence. Q-learning and DQN are in the middle, reflecting their moderate complexity and resource requirements.

These differences reflect the differences in the underlying mechanisms and optimization processes of each algorithm. DDPG and SAC are based on deep reinforcement learning and require frequent updates in complex policy spaces to optimize long-term goals, resulting in high computational overhead and slow convergence. However, they offer better recommendation accuracy and long-term performance, which aligns with the objectives of this paper. CF has a simple structure, relies solely on matrix factorization, consumes few resources, and is suitable for simple tasks, but has weak personalization capabilities. Q-learning and DQN focus on short-term optimization, with relatively

Table 3. Comparison of computational performance of different algorithms

Algorithm	Training Time per Iteration (s)	CPU Usage (%)	CPU Usage (%)	Model Update Frequency (updates/s)	Convergence Time (min)	Memory Usage (MB)
DDPG	1.85	65	80	0.25	35	1024
CF	0.72	30	10	0.1	15	512
Q-learning	1.35	55	40	0.2	30	768
DQN	1.55	60	75	0.22	40	850
SAC	1.6	70	85	0.3	45	950

rough strategy updates, and their efficiency and stability are inferior to those of DDPG and SAC.

3.5. Multi-time-scale Synergy

Reinforcement learning recommendation systems are prone to overfitting due to shortterm behavioral data, neglecting long-term goals. This paper introduces a multi-time-scale reward function, which effectively alleviates the problem of short-term preference overfitting by balancing immediate and long-term rewards.

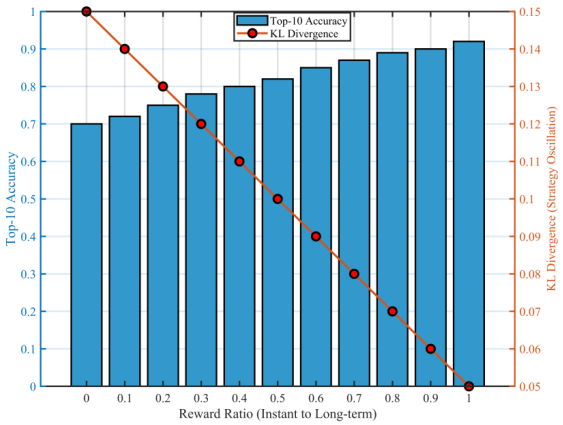

Fig. 7. Analysis of multi-time-scale rewards

Fig. 7 illustrates how multi-time-scale rewards affect the performance of the recommendation system, showing the correlation between Top-10 accuracy and the magnitude of policy fluctuations. The horizontal axis represents the reward ratio, ranging from 0 (biased towards immediate rewards) to 1 (biased towards long-term rewards). The left vertical axis shows Top-10 accuracy. As the reward ratio increases, accuracy gradually increases, indicating that the introduction of long-term goals improves recommendation accuracy. The right vertical axis shows the KL divergence, reflecting the magnitude of policy fluctuations. As the reward ratio gradually shifts from short-term immediate rewards to long-term rewards, the KL divergence decreases, indicating that the volatility of policy updates decreases and the policy becomes more stable. Top-10 accuracy increases from 0.70 to 0.92 , while the KL divergence decreases from 0.15 to 0.05 . This demonstrates that as the reward ratio adjusts, the algorithm can mitigate policy fluctuations while maintaining high accuracy, alleviating the problem of short-term preference overfitting.

Multi-time-scale rewards effectively balance immediate and long-term rewards. The short-term preference in the early stage leads to low accuracy and large strategy fluctuations. With the increase of the long-term reward ratio, the Top-10 accuracy improves, the KL divergence decreases, and the strategy becomes more stable and converges better. It indicates that this algorithm effectively alleviates short-term preference overfitting by reasonably adjusting the reward ratio and taking into account both accuracy and stability.

3.6. State Representation Interpretability

This section explores the optimization effect of adaptive state representation on lowfrequency user recommendations. To address the difficulty of traditional algorithm preference capture caused by the scarcity of interaction data, this method generates precise dynamic state vectors through multimodal feature extraction and self-attention weighted fusion, effectively inferring the potential interests of low-frequency users, and improves the accuracy of recommendations.

Fig. 8 shows the recommendation accuracy for low-frequency and high-frequency users at different interaction times. The horizontal axis represents the number of user interactions, divided into three stages: $\leq 5, 6-10$, and ≥ 11. These stages represent the behavior of lowfrequency and high-frequency users at different data densities. The vertical axis represents recommendation accuracy. The accuracy for low-frequency users ranges from 0.55, 0.60, to 0.68 , while the accuracy for high-frequency users ranges from $0.80, 0.82$, to 0.85 , maintaining a consistently high level. The accuracy of infrequent users gradually improves with increasing interactions, indicating that the algorithm

Table 4. Ablation study results

Variant	Top-10 Acc.	7-day Retention	Cold-start Coverage	KL Divergence
Full Model	0.88	0.85	0.60	0.02
Self-Attention (A)	0.82	0.81	0.57	0.08
GRU (B)	0.85	0.72	0.55	0.05
Long-term Reward (C)	0.86	0.83	0.48	0.04
PER (D)	0.87	0.84	0.59	0.03

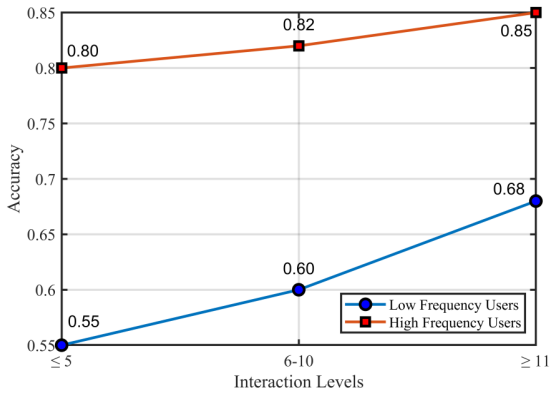

Fig. 8. User adaptation status

is gradually adapting to the user's learning behavior. The accuracy of frequent users is already high and fluctuates less, indicating that they have more historical data, and the system is able to complete optimization earlier.

Adaptive state representation significantly enhances the recommendation effect for lowfrequency users. With a small amount of interaction data, the model can gradually optimize the strategy, accurately model learning habits, and avoid short-term overfitting. However, for highfrequency users, due to the abundance of data, the accuracy rate changes relatively little. This method effectively enhances the learning and recommendation capabilities under the condition of scarce data, demonstrating its adaptability advantage to low-frequency scenarios.

3.7. Ablation Study

We evaluate four variants: (A) replace self-attention with mean pooling; (B) remove GRU, using only current features; (C) set $\lambda = 0$ in reward function (no long-term reward); (D) replace PER with uniform replay.

Results in Table 4 show: removing self-attention (A) drops Top-10 accuracy by 6.2%; disabling GRU (B) reduces 7-day retention to 72%; eliminating long-term reward (C) cuts cold start coverage to 48%; uniform replay (D) slows convergence by 12 minutes. These confirm the necessity of each component.

4. Conclusions

Aiming at the problems of short-term overfitting and long-term strategy instability existing in reinforcement learning in personalized English vocabulary learning, this paper proposes a dual-module collaborative framework that integrates dynamic feature encoders and multi-time-scale strategy networks to achieve decoupled modeling of short-term behaviors and long-term goals. By introducing a time decay factor and dynamically balancing immediate and delayed rewards, the accuracy of recommendations, memory retention rate, and cold start adaptability have been significantly improved. Experiments demonstrate that this method outperforms traditional baseline models such as collaborative filtering and Q-learning in terms of top- N accuracy, strategy stability, and resource efficiency. However, the algorithm still suffers from computational overhead, and the initial recommendation coverage for very lowfrequency users still needs to be improved. Future research can focus on lightweight model design, cross-domain knowledge transfer, and multimodal user feature enhancement to further enhance the generalization and practical application value of personalized learning recommendations.

References

[1] X. Chen, D. Zou, H. Xie, and G. Cheng, (2021) *"Twenty years of personalized language learning"* **Educational Technology & Society** ***24***(1): 205–222.

[2] J. Lahiassi, S. Aammou, and O. Warraki, (2023) *"Enhancing personalized learning with a recommendation system in private online courses"* **Conhecimento & Diversidade** ***15***(39): 176–189. DOI: 10.18316/rcd.v15i39.11144.

[3] J.-W. Tzeng, N.-F. Huang, A.-C. Chuang, T.-W. Huang, and H.-Y. Chang, (2025) *"Massive open online course recommendation system based on a reinforcement learning algorithm"* **Neural Computing and Applications** ***37***(18): 11607–11618. DOI: 10.1007/s00521-023-08686-8.

[4] Y. Deng, Y. Li, B. Ding, and W. Lam, (2022) *"Leveraging long short-term user preference in conversational recommendation via multi-agent reinforcement learning"* **IEEE Transactions on Knowledge and Data Engineering** ***35***(11): 11541–11555. DOI: 10.1109/TKDE.2022.3225109.

[5] T. Liu, Q. Wu, L. Chang, and T. Gu, (2022) *"A review of deep learning-based recommender system in e-learning environments"* **Artificial Intelligence Review** ***55***(8): 5953–5980. DOI: 10.1007/s10462-022-10135-2.

[6] J. Lahiassi, S. Aammou, and Y. Jdidou, (2025) *"PERSONALIZED LEARNER RECOMMENDATIONS: ENHANCING GROUP DYNAMICS IN COLLABORATIVE LEARNING"* **Conhecimento & Diversidade** ***17***(45): 591–607. DOI: 10.18316/rcd.v17i45.12504.

[7] R. Fariani, K. Junus, and H. Santoso, (2023) *"A systematic literature review on personalised learning in the higher education context"* **Technology, Knowledge and Learning** ***28***(2): 449–476. DOI: 10.1007/s10758-022-09628-4.

[8] J. Luo, F. Li, and J. Jiao, (2025) *"A dynamic multiobjective recommendation method based on soft actor-critic with discrete actions"* **Journal of King Saud University Computer and Information Sciences** ***37***(1): 1. DOI: 10.1007/s44443-025-00016-3.

[9] S. Cheng, Z. Wu, M. Qian, and W. Huang, (2024) *"Point-of-interest recommendation based on bidirectional self-attention mechanism by fusing spatio-temporal preference"* **Multimedia Tools and Applications** ***83***(9): 26333–26347. DOI: 10.1007/s11042-023-16542-z.

[10] H. Wang and W. Fu, (2021) *"Personalized learning resource recommendation method based on dynamic collaborative filtering"* **Mobile Networks and Applications** ***26***(1): 473–487. DOI: 10.1007/s11036-020-01673-6.

[11] X. Shi, Q. Liu, H. Xie, Y. Bai, and M. Shang, (2024) *"Maximum entropy policy for long-term fairness in interactive recommender systems"* **IEEE Transactions on Services Computing** ***17***(3): 1029–1043. DOI: 10.1109/TSC.2024.3349636.

[12] N. Vedavathi and K. Anil Kumar, (2021) *"An efficient e-learning recommendation system for user preferences using hybrid optimization algorithm"* **Soft Computing** ***25***(14): 9377–9388. DOI: 10.1007/s00500-021-05753-x.

[13] Z. Li and H. Wang, (2023) *"Study on recommendation of personalized learning resources based on deep reinforcement learning"* **International Journal of Information and Communication Technology** ***23***(4): 299–313. DOI: 10.1504/IJICT.2023.134832.

[14] Y. Lin, Y. Liu, F. Lin, L. Zou, P. Wu, W. Zeng, H. Chen, and C. Miao, (2023) *"A survey on reinforcement learning for recommender systems"* **IEEE Transactions on Neural Networks and Learning Systems** ***35***(10): 13164–13184. DOI: 10.1109/TNNLS.2023.3280161.

[15] M. Mu and M. Yuan, (2024) *"Research on a personalized learning path recommendation system based on cognitive graph with a cognitive graph"* **Interactive Learning Environments** ***32***(8): 4237–4255. DOI: 10.1080/10494820.2023.2195446.

[16] R. Guan, H. Pang, F. Giunchiglia, Y. Liang, and X. Feng, (2022) *"Cross-domain meta-learner for cold-start recommendation"* **IEEE Transactions on Knowledge and Data Engineering** ***35***(8): 7829–7843. DOI: 10.1109/TKDE.2022.3208005.

[17] J. Meng, (2025) *"AVAR-RL: Adaptive reinforcement learning approach for personalized English vocabulary acquisition"* **Discover Artificial Intelligence** ***5***(1): 1–17. DOI: 10.1007/s44163-025-00584-3.

[18] D. Chen, S. Yongchareon, E.-K. Lai, J. Yu, Q. Sheng, and Y. Li, (2022) *"Transformer with bidirectional GRU for nonintrusive, sensor-based activity recognition in a multiresident environment"* **IEEE Internet of Things Journal** ***9***(23): 23716–23727. DOI: 10.1109/JIOT.2022.3190307.

[19] M. Afsar, T. Crump, and B. Far, (2022) *"Reinforcement learning based recommender systems: A survey"* **ACM Computing Surveys** ***55***(7): 1–38. DOI: 10.1145/3543846.

[20] L. Xia, C. Huang, Y. Xu, and J. Pei, (2022) *"Multi-behavior sequential recommendation with temporal graph transformer"* **IEEE Transactions on Knowledge and Data Engineering** ***35***(6): 6099–6112. DOI: 10.1109/TKDE.2022.3175094.

[21] R. Zhang, D. Zou, and H. Xie, (2022) *"Spaced repetition for authentic mobile-assisted word learning: Nature, learner perceptions, and factors leading to positive perceptions"* **Computer Assisted Language Learning** ***35***(9): 2593–2626. DOI: 10.1080/09588221.2021.1888752.

[22] M. Walsh, M. Krusmark, T. Jastrembski, D. Hansen, K. Honn, and G. Gunzelmann, (2023) *"Enhancing learning and retention through the distribution of practice repetitions across multiple sessions"* **Memory & Cognition** ***51***(2): 455–472. DOI: 10.3758/s13421-022-01361-8.

[23] X. Liu, M. Yu, C. Yang, L. Zhou, H. Wang, and H. Zhou, (2024) *"Value distribution DDPG with dual-prioritized experience replay for coordinated control of coal-fired power generation systems"* **IEEE Transactions on Industrial Informatics *20***(6): 8181–8194. DOI: 10.1109/TII.2024.3369712.

[24] M. Daniel, A. Magassouba, M. Aranda, J. Ramón, R. Rodriguez, and Y. Mezouar, (2023) *"Multi actor-critic DDPG for robot action space decomposition: A framework to control large 3D deformation of soft linear objects"* **IEEE Robotics and Automation Letters *9***(2): 1318–1325. DOI: 10.1109/LRA.2023.3342672.

[25] D. Dutta and S. Upreti, (2022) *"A survey and comparative evaluation of actor-critic methods in process control"* **The Canadian Journal of Chemical Engineering *100***(9): 2028–2056. DOI: 10.1002/cjce.24508.

[26] A. Fakhrezi, G. Budiman, and D. Perdana, (2025) *"Enhancing Soybean Fertilization Optimization with Prioritized Experience Replay and Noisy Networks in Deep Q-Networks"* **Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika (JITEKI) *11***(2): 154–168. DOI: 10.26555/jiteki.v11i2.30690.

[27] D. Song, T. Ma, J. Shen, and F. Xu, (2024) *"Incremental learning-based quantitative crack detection using prioritized experience replaying and layered importance sampling"* **IEEE Sensors Journal *24***(15): 25132–25140. DOI: 10.1109/JSEN.2024.3415810.

[28] Z. Lou, Y. Wang, S. Shan, K. Zhang, and H. Wei, (2024) *"Balanced prioritized experience replay in off-policy reinforcement learning"* **Neural Computing and Applications *36***(25): 15721–15737. DOI: 10.1007/s00521-024-09913-6.

[29] V. Coscrato and D. Bridge, (2023) *"Estimating and evaluating the uncertainty of rating predictions and top-n recommendations in recommender systems"* **ACM Transactions on Recommender Systems *1***(2): 1–34. DOI: 10.1145/3584021.

[30] W. Zhao, Z. Lin, Z. Feng, P. Wang, and J.-R. Wen, (2022) *"A revisiting study of appropriate offline evaluation for top-N recommendation algorithms"* **ACM Transactions on Information Systems *41***(2): 1–41. DOI: 10.1145/3545796.

[31] F. Nielsen, (2022) *"The Kullback–Leibler divergence between lattice Gaussian distributions"* **Journal of the Indian Institute of Science *102***(4): 1177–1188. DOI: 10.1007/s41745-021-00279-5.

[32] M. Rahad, R. Shabab, M. Ahammad, M. Reza, A. Karmaker, and M. Hossain, (2025) *"KL-FedDis: A federated learning approach with distribution information sharing using Kullback-Leibler divergence for non-IID data"* **Neuroscience Informatics *5***(1): 100182. DOI: 10.1016/j.neuri.2024.100182.