BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 74(4), 2026, Article number: e158770
DOI: 10.24425/bpasts.2026.158770



# DouRD: enhancing DouDizhu AI with role-differentiated modeling

**Yuezhongyi SUN and Shouzhen ZHANG***

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China

**Abstract.** Imperfect information games impose greater demands on AI decision-making than perfect information settings, requiring models to infer hidden information, reason about opponent strategies, and dynamically optimize policies under uncertainty. In this study, we proposed a novel role-differentiated modeling approach within the deep Monte Carlo framework to enhance DouDizhu AI, the challenging three-player asymmetric imperfect information game. Our method incorporated attention mechanisms with role-specific adaptations to investigate their differential impacts on the landlord and peasant roles. Key findings demonstrate that: (1) the landlord and peasant roles require fundamentally distinct model architectures: experiments confirmed their functional independence; (2) attention mechanisms exhibit role-dependent effectiveness: CBAM significantly improved Peasant strategy execution, whereas SE and ECA offered moderate gains, while Self-Attention showed no enhancement; (3) surprisingly, applying attention mechanisms to the landlord role led to performance degradation, reinforcing the superiority of LSTM for this role. These results highlight the importance of role-aware architecture design in the imperfect information game setting, and challenge the universal applicability of attention mechanisms.

**Keywords:** attention mechanism; deep Monte Carlo; DouDizhu; imperfect information game.

## 1. INTRODUCTION

Machine gaming has emerged as a fundamental experimental platform in artificial intelligence research, providing an ideal testbed for optimizing agent decision-making, strategy learning and human-computer interaction. In recent years, AI has achieved remarkable advancements in this domain, spanning board games, poker games, real-time strategy games and multiplayer online competitive games. In board games, Deep Blue leveraged the Alpha-beta pruning algorithm to defeat world champion Garri Kasparov, while subsequent agents such as AlphaZero [1] further revolutionized the field. In Go, AlphaGo [2] integrated deep reinforcement learning with Monte Carlo tree search (MCTS) [3] to defeat world champion Lee Sedol. Its successor, AlphaGo Zero [4] perfected this approach by achieving fully autonomous learning, surpassing human experts without prior gameplay data. In poker, agents such as DeepStack [5], Libratus [6] and Pluribus [7] demonstrated exceptional strategic reasoning and equilibrium computation capabilities in both two-player and multiplayer Texas hold'em. Notably, Pluribus became the first AI to defeat top human professionals in multiplayer Texas hold'em, marking a milestone in imperfect information gaming. In video games, the Arnold [8] agent, integrating Deep Q-Network (DQN) [9] and long short-term memory (LSTM), secured victory in the VizDoom competition. AlphaStar [10], utilizing a combination of imitation learning and reinforcement learning, attained a skill level comparable to top professional players in StarCraft II, while OpenAI Five [11] successfully defeated the world champion team OG in Dota2, demonstrating the adaptability of AI in highly complex multiplayer gaming environments.

DouDizhu, a representative imperfect information game, features a vast and complex action space due to combinations of cards [12]. Compared to poker games such as Texas hold'em, its action space presents greater challenges for abstraction. As a result, reinforcement learning methods such as DQN and Asynchronous Advantage Actor-Critic (A3C) [13] have exhibited limited effectiveness in DouDizhu. Research by You *et al.* [14] has shown that these approaches fail to achieve satisfactory strategic performance in this environment.

To address this challenge, DouZero [15] combines deep learning-based feature extraction with Monte Carlo-based strategy optimization, providing an effective solution for DouDizhu agents. However, conventional approaches typically employ the same LSTM architecture for both landlords and peasants, overlooking their distinct decision-making needs.

To evaluate the impact of different model architectures on agent strategy learning, this study integrates four attention mechanisms into the DouZero framework. Leveraging the heterogeneous objectives of roles in DouDizhu, a role-differentiated architecture is introduced: the landlord agent preserves the LSTM module to support long-term strategic planning, while peasant agents employ attention mechanisms to improve inference of hidden cards held by the landlord. Experimental results demonstrate that this role-aware modeling approach enhances overall strategic performance without introducing parameter interference.

The primary contributions of this work are as follows:

1) four well-established attention mechanisms are systematically introduced and analyzed in the context of DouDizhu,

*e-mail: 2320410119@stu.hrbust.edu.cn

Manuscript submitted 2025-06-14, revised 2026-03-11, initially accepted for publication 2026-04-02, published in July 2026.

demonstrating their effectiveness in feature enhancement for imperfect-information games.
2) a role-differentiated modeling approach is proposed based on the asymmetry between landlord and peasant objectives, enabling more targeted strategy optimization.
3) the DouZero training pipeline is extended, and extensive experiments show consistent improvements in both win rate (WP) and average difference in points (ADP).

## 2. RELATED WORK

DouDizhu, characterized by imperfect information, asymmetric roles and multi-agent interactions, has recently become a prominent testbed for reinforcement learning research. Extensive studies have been conducted from various perspectives, including strategy optimization, model architecture and search methods, contributing to the continuous advancement of DouDizhu AI.

Early research primarily focused on architectural design and the enhancement of strategy generalization. For instance, Conditional Q-Network (CQN) [14] formulated the card-playing task as a conditional Q-learning problem, introducing prior constraints into the reinforcement learning framework, which effectively improved policy rationality and controllability. Supervised learning (SL) methods leveraged expert gameplay data to perform supervised training, providing an initial policy for reinforcement learning and thereby improving convergence efficiency in the early learning stages. To enhance policy robustness and generalization, RHCP-v2 [16] integrated rule-based heuristics and probability adjustment mechanisms, leading to improved model performance across diverse gameplay scenarios. In addition, DeltaDou [17] addressed the issues of action compression and offensive-defensive strategy switching by introducing a compact action representation and a switchable policy module, making the learning process more efficient and the gameplay strategies more adaptive.

Building on these advancements, DouZero introduced the first end-to-end DouDizhu AI system, eliminating the need for search and instead leveraging deep neural networks combined with Monte Carlo returns for policy optimization. This approach significantly improved performance in terms of both win rate and score difference. Subsequently, several extended versions further propelled the development of this framework.

DouZero+ [18] extends DouZero by incorporating opponent modeling and coach-guided learning, further enhancing the inference capability and training efficiency of the model. The opponent modeling module learns from gameplay history to estimate the probability distribution of the hidden hands of opponents, while the coach-guided mechanism filters high-value opening data to facilitate more effective strategy learning. Experimental results indicate that DouZero+ improves ADP by 0.3, underscoring its enhanced strategy optimization capability. However, this method exhibits a strong dependence on labeled data, resulting in higher training costs.

DouRN [19] refines DouZero by integrating residual networks and a bidding mechanism to enhance representational power and optimize opening strategies. The residual network mitigates the vanishing gradient problem, thereby improving training stability and reinforcing strategy learning. The bidding mechanism targets the optimization of the top 10% of critical decisions, further refining strategy quality. Experimental results demonstrate that compared to DouZero, DouRN achieves a 7% increase in win rate.

Unlike DouZero+, which primarily relies on high-value opening data, Qiao *et al.* [20] proposed a method integrating minimum combination search (MCS) with opponent modeling. This approach improves the training efficiency of DouZero and enhances strategy generation. MCS reduces redundant search space, thereby improving search efficiency, while opponent modeling augments strategy adaptability within gameplay. Experimental results reveal that this approach reduces training time by 74.5% while maintaining performance comparable to the original model. Furthermore, this study extends the DMC method to the Big2 game.

In summary, existing research on DouDizhu AI has primarily focused on optimizing architectural frameworks and improving the efficiency of strategy search. For instance, DouZero promotes policy convergence via Monte Carlo methods, while DouZero+ and DouRN enhance training stability and strategic performance through the incorporation of opponent modeling, residual networks, and bidding mechanisms. However, these approaches largely neglect the direct influence of feature modeling techniques on decision quality and typically employ identical network architectures for both landlord and peasant agents, without accounting for the impact of role-specific differences on strategic behavior. To address this limitation, this study introduces attention mechanisms and proposes a role-differentiated modeling approach to enhance the ability of the model to perceive hidden information and adapt to complex game environments, thereby advancing research on modeling optimization for DouDizhu AI.

## 3. PRELIMINARY STAGE

In this study, we built a DouDizhu AI based on the DouZero framework and explored different information modeling methods to optimize the decision-making capability of the agent. To clearly describe the methodology, this section first introduces the basic rules of the DouDizhu game, followed by an explanation of the core principles of deep Monte Carlo (DMC) and the encoding methods for states and actions in DouDizhu AI. The subsequent sections will provide a detailed discussion of role-differentiated modeling and the integration of attention mechanisms.

### 3.1. Game rules of DouDizhu

DouDizhu is a widely played three-player poker game in China, using a standard 54-card deck that includes two Jokers. The role of the "landlord" is assigned through a bidding phase, where the player with the highest bid receives an additional three bottom cards and assumes the landlord position. The remaining two players form a temporary alliance, referred to as the "peasants (farmers)", and work together to defeat the landlord. Players take turns in clockwise order, and legal combinations include Solo, Pair, Trio, Bomb and other patterns, as illustrated in Fig. 1.

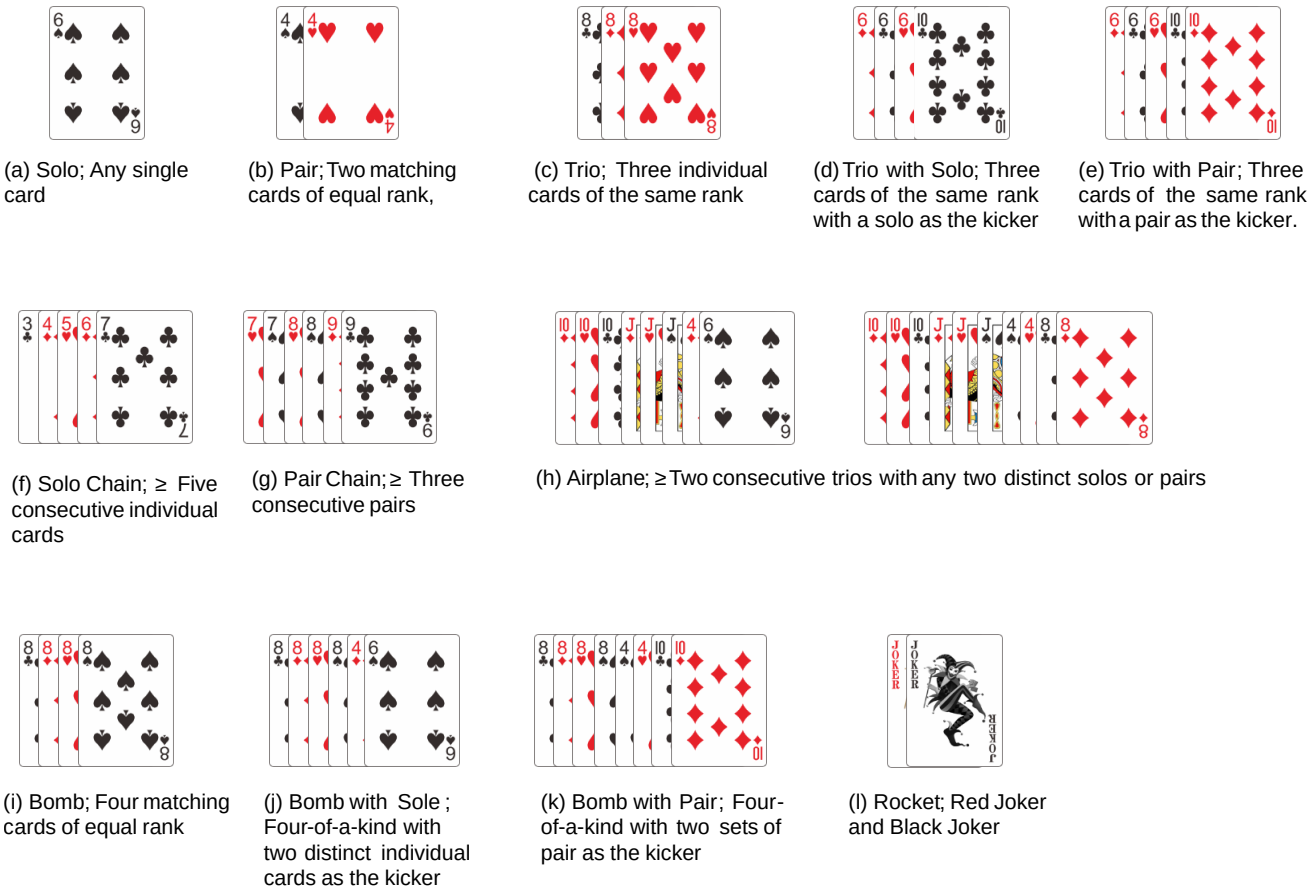


**Fig. 1.** Types of card combinations. Except for combinations (i) and (l), each combination can only defeat others of the same type. Combination (l) can also defeat combination (i)

The win condition is as follows: if the landlord plays all cards first, the landlord wins; otherwise, if either peasant plays all cards first, the peasant alliance wins. DouDizhu is characterized by several complex properties, including imperfect information, a large action space, asymmetric player roles, and the coexistence of short-term tactics and long-term strategic planning. These characteristics make it a valuable research platform for reinforcement learning and multi-agent decision-making.

### 3.2. Deep Monte Carlo method

Monte Carlo methods are a class of sampling-based reinforcement learning techniques whose core idea is to optimize policies by estimating the cumulative returns of states or state-action pairs based on multiple complete episodes. Specifically, Monte Carlo methods compute the cumulative return for each state-action pair at the end of an episode and use this return to update the Q-value, allowing AI to make better decisions under the same state. However, traditional Monte Carlo methods update policies only after an episode is completed, resulting in slow convergence, low data efficiency and high estimation variance due to the dependence on entire episode outcomes.

To address these issues, DouZero introduced the deep Monte Carlo (DMC) method, which enhances data efficiency by incorporating temporal difference learning and accelerates training through parallel self-play. DMC employs a Q-network (such as LSTM + MLP) to learn the state-action value function, eliminating the need to store every state- action pair individually, thereby improving policy generalization. Furthermore, DMC utilizes multiple agents performing parallel self-play for data sampling, effectively enhancing the training efficiency and decision-making stability of DouDizhu AI.

In the DouDizhu task, DMC generates a large amount of training data through self-play, where the agent makes decisions based on the current policy and records the game state, actions and final rewards. At the end of each game, the system computes the cumulative return for each state-action pair using the Monte Carlo method and optimizes the Q-network via gradient descent, allowing AI to make better decisions under similar states. This iterative training process continuously improves the strategic performance of the agent through repeated gameplay and self-optimization. The specific procedure of DMC is as follows:

1) initialize a randomly generated policy.
2) for all state-action pairs in the current game, update the corresponding Q-network using the MSE loss function.
3) repeat steps 1–2, continuously conducting self-play training until the policy converges or the predefined training iterations are reached.

### 3.3. State and action encoding

In DouDizhu AI training, an appropriate state and action encoding scheme is crucial for the learning effectiveness of the model. This study adopts the encoding approach used in the DouZero framework to ensure consistency with the original DMC method while providing a stable foundation for subsequent role-differentiated modeling.

In DouZero, a $4 \times 15$ one-hot matrix is used to encode the hand cards (as illustrated in Fig. 2). Since DouDizhu does not differentiate between suits, the 15 columns of the matrix correspond to numerical values from 1 to 13 for standard cards, along with two additional columns representing the Joker cards. The four rows indicate the specific count of each card in hand. Since the Jokers exist as single cards, there are six fixed zero positions in the $4 \times 15$ one-hot matrix. To optimize storage and computation, this representation can further be mapped to a $1 \times 54$ one-hot vector, where each index uniquely corresponds to the presence of a specific card.

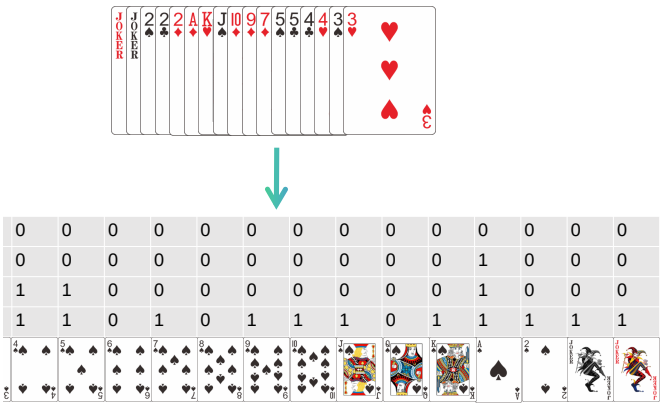

**Fig. 2.** Example card encoding

**Table 1**

Player Features. In DouDizhu, four identical number cards or Jokers form a bomb, with a total of 15 types. A round is defined as a sequence in which all three players take one action

| | Feature | Size |
|---|---|---|
| Action | Card matrix of the action | 54 |
| State | Card matrix of the hand cards | 54 |
| | Card matrix of the union of the hands of other players | 54 |
| | Card matrix of the action of the other player 1 | 54 |
| | Card matrix of the action of the other player 2 | 54 |
| | One-hot vector representing number of cards left for player 1 | 17 |
| | One-hot vector representing number of cards left for player 2 | 17 |
| | One-hot vector representing the current number of bombs | 15 |
| | Concatenated matrix of the most recent 5 rounds | $5 \times 162$ |

Beyond hand card information, state encoding includes additional key features to enhance the understanding of the game situation by the AI. These features comprise the number of remaining cards for all players, the current usage of bombs, and the recent move history, enabling the agent to make more accurate predictions about the strategies of opponents and to refine its own decision-making process. Similarly, actions are also encoded in one-hot format to standardize input data and facilitate neural network computation. Table 1 provides a detailed description of these state encoding features.

## 4. METHOD

### 4.1. Problem formulation

As a game environment characterized by role asymmetry, partial observability and dynamic strategic interactions, DouDizhu presents core challenges in the coordinated design of state representation, strategy optimization objectives and role-differentiated learning approaches.

In this study, the DouDizhu task is formulated as a multi-agent reinforcement learning (MARL) problem, where each agent – either a landlord or a peasant – makes decisions based on partially observable information, aiming to maximize individual or alliance-level win rates within a limited number of rounds. The modeling framework is described as follows:

- state space: each agent observes partial information, including its own hand cards, the history of played cards, the remaining number of cards held by opponents, the number of bombs used, and other contextual features (see Section III.C for details).
- action space: following the representation method used in DouZero, all legal actions are encoded as 54-dimensional one-hot vectors.
- reward function: agents receive win-based rewards (+1 for wins/−1 for losses) for end-result optimization, or score-based rewards (continuous values following DouDizhu scoring rules, e.g. bomb multipliers) for granular policy guidance.
- policy modeling: the landlord utilizes a structure suited for long-term dependency modeling, whereas the peasants emphasize local cooperative feature learning. While agents share network parameters, differentiation is achieved via role-specific tags.

In contrast to the unified policy structure used in DouZero, this study proposes a role-differentiated modeling approach enhanced with attention mechanisms. These improvements enhance the collaborative decision-making capability of peasant agents and boost the capacity for global strategic awareness of the landlord, achieving better alignment with the heterogeneous objectives of different roles in the DouDizhu environment.

### 4.2. Differentiated modeling of DMC framework

In the DouDizhu task, the strategic objectives and decision-making processes of the landlord and peasants differ significantly. The landlord aims to maximize the efficiency of card playing and develop long-term strategies, whereas the peasants

must collaborate to infer the hand of the landlord and restrict their plays to increase their chances of winning.

Therefore, directly applying a unified information modeling approach may fail to fully leverage the strengths of each role. The objective of this section is to propose a role-differentiated DMC training framework that optimizes the strategies of the landlord and peasants in alignment with their respective decision-making needs.

On the landlord side, we adopt the LSTM structure from DouZero (as illustrated in Fig. 3). This structure models sequential dependencies through recursive processing, capturing key patterns in the history of played cards while incorporating current state information to refine Q-value estimation. The input to LSTM is formatted as (batch_size, seq_len, 162), where batch_size represents the sample batch size, seq_len denotes the length of the historical move sequence, and the 162-dimensional feature vector encodes the moves of all three players. Each move of a player is represented by a 54-dimensional one-hot vector. The output takes the form of the moves of all three players. Each move of a player is represented by a 54-dimensional one-hot vector. The output is a 128-dimensional hidden state, which is subsequently processed by an MLP layer for decision optimization. By leveraging the capability of LSTM to model long-term dependencies, the landlord can plan card plays over an extended time horizon, thereby improving overall efficiency of card playing.

Unlike the landlord side, the decision-making process on the peasant side focuses more on modeling local key information (as illustrated in Fig. 4). Therefore, we replace LSTM with attention mechanisms to enhance the ability of the model to extract critical features. Attention mechanisms adaptively assign weights to different state variables, allowing the model to focus more precisely on information that has a greater impact on the current decision. In our experiments, we evaluate four attention mechanisms: CBAM (convolutional block attention module) [21], SE (squeeze-and-excitation) [22], ECA (efficient channel attention) [23] and self-attention [24]. However, the original designs of these mechanisms are primarily intended for image processing and NLP tasks, making their input formats inherently different from the feature representations in the DouDizhu task. To integrate these mechanisms into the DMC training framework, we make necessary modifications to align them with the structural requirements of DouDizhu AI.

CBAM integrates channel attention and spatial attention modules to precisely model key features. In this task, the channel attention module first computes the importance of each feature dimension in the input (batch_size, seq_len, 162), followed by the spatial attention module, which further filters critical information. The output dimension of CBAM remains identical to the input, i.e. (batch_size, seq_len, 162). A fully connected layer is then applied for dimensionality reduction to ensure compatibility with the LSTM output.

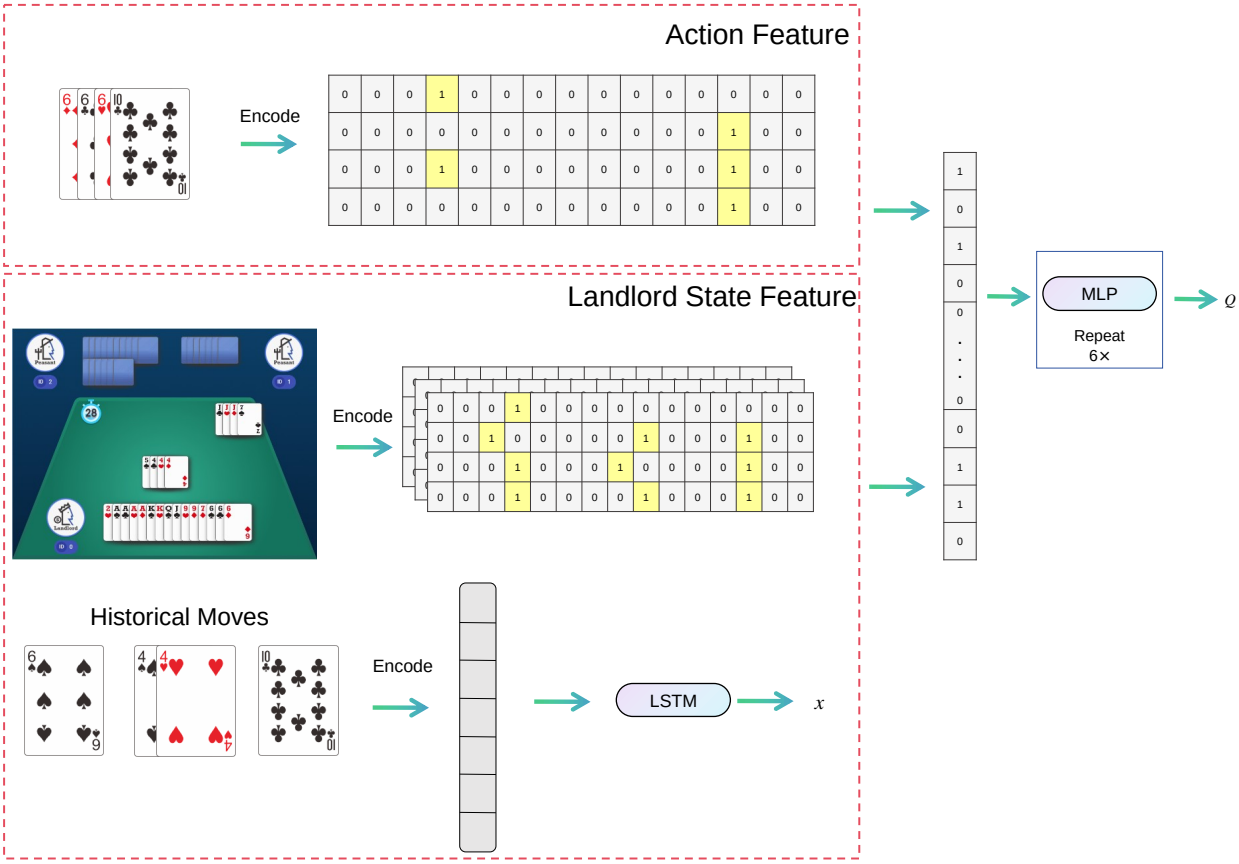


**Fig. 3.** Q-network framework for the landlord role

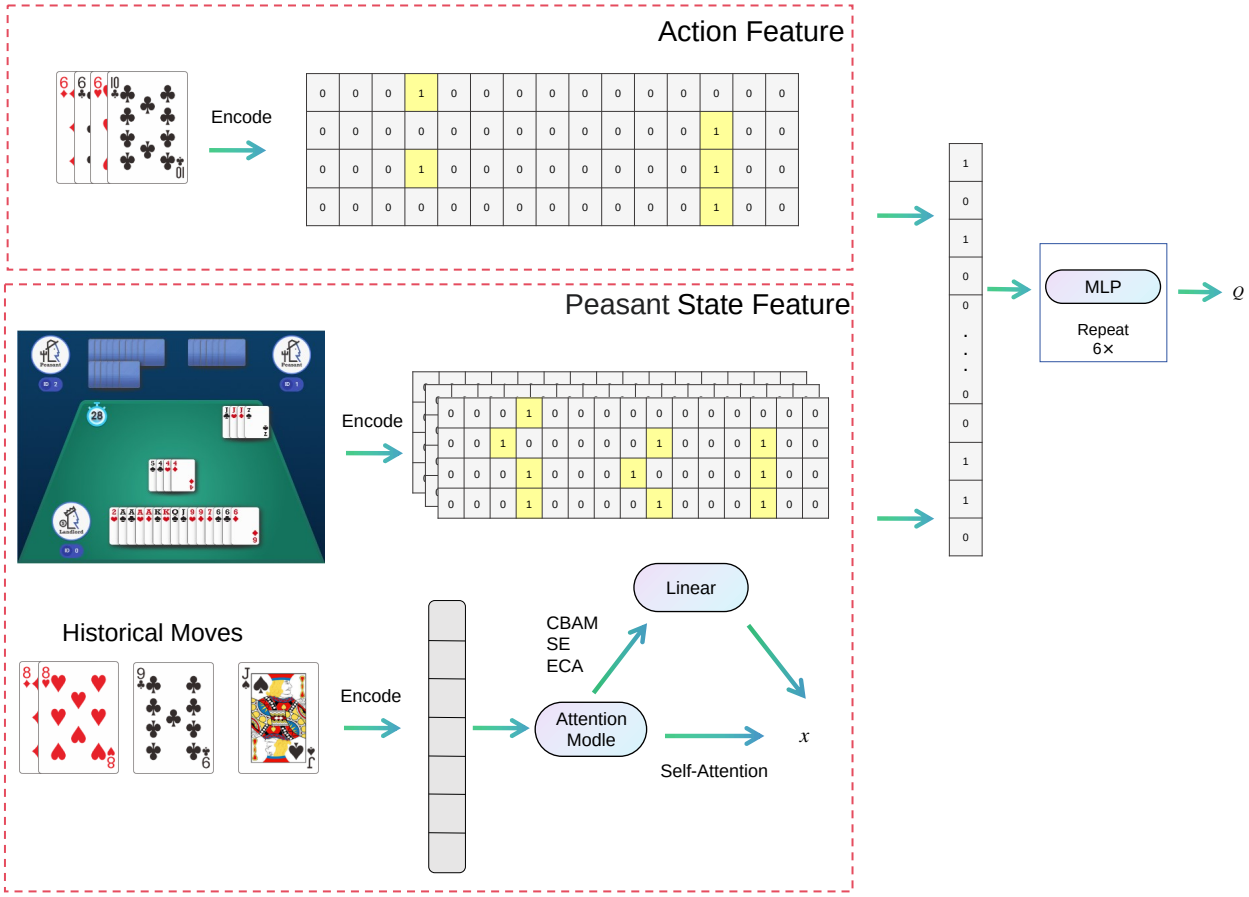


**Fig. 4.** Q-network framework for the peasant role. Self-attention does not require a linear layer, whereas CBAM, SE and ECA do

SE employs global average pooling (GAP) to compute channel-wise attention weights, allowing the model to adaptively emphasize the most discriminative features. SE focuses solely on channel-wise attention, taking an input of (batch_size, seq_len, 162). After global pooling, the features are compressed into (batch_size, seq_len, 1), then passed through a two-layer fully connected network to adjust the weights and map them back to (batch_size, seq_len, 162). Finally, a fully connected layer is used for dimensionality reduction.

ECA applies 1D convolution to capture local channel dependencies, eliminating the need for global pooling while effectively extracting feature correlations. ECA operates directly on (batch_size, seq_len, 162) using 1D convolution, learning the relative importance of different channels and adjusting the input accordingly. The output dimension remains unchanged, and a fully connected layer is subsequently used for dimensionality reduction.

Self-attention in turn models global dependencies, helping the peasant agent better infer the card-playing strategy of the landlord. The input to self-attention is (batch_size, seq_len, 162), and the computation involves mapping query (Q), key (K) and value (V) vectors, and calculating attention scores. Since self-attention produces a variable output dimension, we adjust the number of heads and hidden dimensions to ensure compatibility with the LSTM structure.

Within the DMC training framework, we retain the multi-process parallel training approach of DouZero, where the agent engages in self-play within the DouDizhu environment, collecting a large volume of game data. The Monte Carlo method is then used to compute cumulative returns, optimizing the Q-network. To evaluate the impact of role- differentiated modeling on performance, we compare the traditional LSTM structure with different attention mechanisms under the same DMC training pipeline. Experimental results demonstrate that incorporating attention mechanisms significantly enhances the ability of the peasant model to interpret the game state, thereby improving overall win rates.

### 4.3. Attention mechanisms

Attention mechanisms are a dynamic feature of the modeling approach that allocate computational resources based on the importance of input features, enhancing the representation of critical information while reducing interference from irrelevant features. In game-playing tasks, attention mechanisms improve the predictive ability and decision adaptability of policy models while enhancing generalization performance [25,26]. Therefore, this study introduces four attention mechanisms – CBAM, SE, ECA and self-attention – and analyzes their applicability to the DouDizhu AI task. To facilitate clearer understanding of the employed mechanisms, Fig. 5 through Fig. 8 illustrate the structural

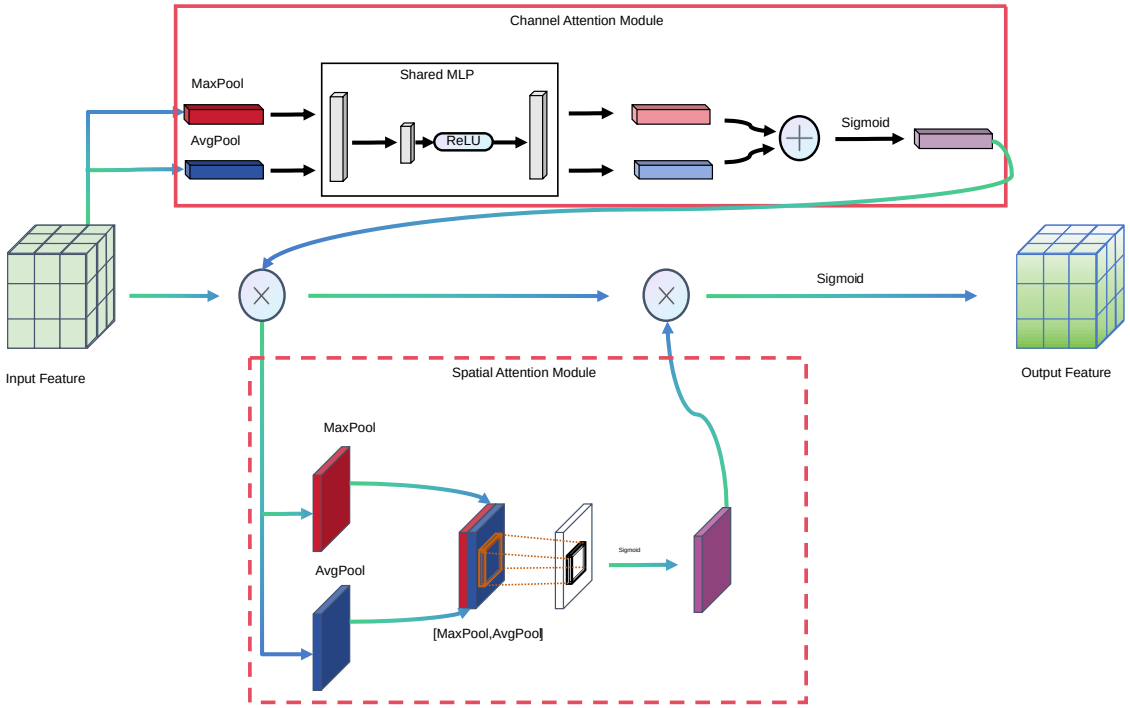


**Fig. 5.** CBAM attention module framework

diagrams of the CBAM, SE, ECA and self-attention modules. All diagrams were reproduced and rendered by the authors based on their practical implementations. Each module was integrated into the DouDizhu AI model in the subsequent experiments to assess the impact of different attention mechanisms on strategic decision-making performance.

CBAM is a dual-attention mechanism that combines channel attention and spatial attention to adaptively enhance the perception of the model regarding critical features. As illustrated in Fig. 5, the CBAM structure consists of two sequential submodules: the channel attention module and the spatial attention module.

In the context of the DouDizhu task, CBAM significantly improves discriminative capability of the model. Channel attention helps highlight strategy-related dimensions that are highly relevant to decision-making, such as current hand strength and opponent behavioral patterns. Spatial attention emphasizes the expression of key action sequences within local contexts, such as card-playing preferences in specific scenarios and sensitivity to changes in the game state.

CBAM workflow proceeds as follows:

1) channel attention module: the input feature map undergoes global average pooling and max pooling to generate two channel descriptor vectors, which are then passed through a shared MLP to produce the channel attention map. Channel attention weights are generated and applied to the input feature map.
2) the channel attention map is used to reweight the input feature map along the channel dimension.
3) spatial attention module: the reweighted feature map is subjected to average pooling and max pooling along the channel dimension. The two resulting spatial descriptors are concatenated and passed through a convolution layer to generate the spatial attention map.
4) the spatial attention map is then used to reweight the feature map along the spatial dimension, producing the final enhanced output.

Through this mechanism, CBAM significantly improves the ability of the model to respond to key information within the input state, thereby enhancing the optimization of card-playing strategies. Experimental results further validate its effectiveness in the DouDizhu task.

SE mechanism is a channel-focused attention method that adaptively recalibrates channel-wise feature responses by modeling the importance relationships among channels in the feature map. Its core idea is to dynamically enhance discriminative channels using global contextual information, thereby improving sensitivity of the model to key features. The structural diagram of the SE module is shown in Fig. 6.

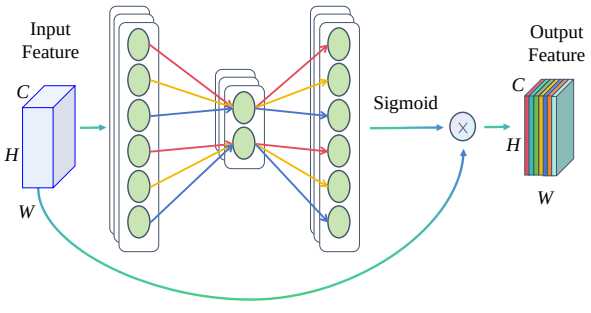


**Fig. 6.** SE attention module framework

In the DouDizhu task, the SE module effectively enhances the channel-wise feature representation in the strategy network, enabling agents to more accurately recognize shifts in the game state and identify critical card-playing features. This contributes to improved strategy stability and decision-making robustness.

SE workflow proceeds as follows:

1) squeeze phase: global average pooling is applied across the spatial dimension of the input feature map, producing a global descriptor vector for each channel.
2) excitation phase: a bottleneck structure consisting of two fully connected layers performs a nonlinear transformation on the channel descriptor vector, generating channel attention weights.
3) these weights are used to reweight the original input feature map along the channel dimension, producing an enhanced feature map.

SE structure is lightweight and easy to implement, rendering it well-suited for feature enhancement in strategic game tasks like DouDizhu. It is particularly effective in emphasizing semantic importance across channels, the strategic awareness and generalization performance of the model.

ECA mechanism is a lightweight enhancement of the SE module, designed to model inter-channel dependencies through local channel interactions. It achieves comparable attention performance while significantly reducing computational complexity. A structural diagram of the ECA module is shown in Fig. 7. Unlike SE, ECA removes fully connected layers to avoid information compression and potential feature loss. Instead, it uses one-dimensional convolution operations to directly capture local dependencies among channels, thereby improving responsiveness of the model to key features and enhancing the efficiency of inter-channel representation.

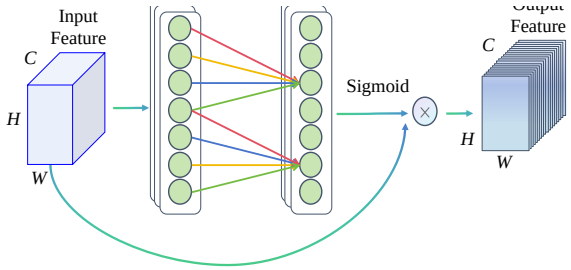


**Fig. 7.** ECA module framework

In the context of the DouDizhu task, the efficiency of ECA makes it particularly suitable for scenarios with limited training resources or strict inference efficiency requirements. While reducing parameter size and improving computational efficiency, ECA still effectively enhances the strategy evaluation capabilities and gameplay performance of the model.

ECA workflow proceeds as follows:

1) channel feature extraction: global average pooling is applied to the input feature map to obtain statistical information for each channel.
2) local dependency modeling: a one-dimensional convolution with a small kernel size is performed on the channel vector to capture local channel relationships.
3) weight generation: the convolution output is passed through a sigmoid activation function to produce channel attention weights.
4) feature enhancement: the original feature map is multiplied channel-wise with the attention weights to generate the final weighted output.
5) ECA offers an excellent balance between efficiency and expressiveness. In the complex decision-making environment of DouDizhu AI, it serves as an ideal complement to the strategy network for effective feature modeling.

Self-attention is a mechanism for modeling global dependencies and it is widely applied in tasks such as natural language processing and image understanding. It is particularly well-suited for scenarios requiring cross-time-step reasoning or capturing long-range feature dependencies. The structural diagram of the self-attention module is shown in Fig. 8.

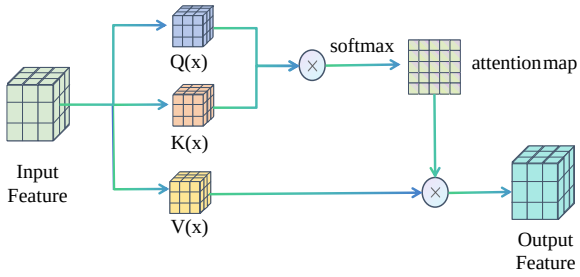


**Fig. 8.** Self-attention module framework

In the DouDizhu AI task, self-attention enhances the perception of the agent regarding the global game state, offering significant advantages for peasant agents when inferring the hidden cards of the landlord and identifying card-playing trends. Unlike traditional LSTM structures that rely on recursive information propagation, self-attention allows for parallel processing of input sequences and directly models dependencies between all positions, thereby improving strategy consistency and decision accuracy over long sequences.

Self-attention workflow proceeds as follows:

1) linear transformation and feature mapping: the input features are projected into query, key and value vectors through three trainable weight matrices.
2) similarity computation: the dot-product similarity between each query and all keys is calculated to form the attention score matrix.
3) weight normalization: the attention scores are normalized using the Softmax function to obtain attention weights.
4) context aggregation: the attention weights are used to compute a weighted sum of the value vectors, resulting in a context representation for each time step.
5) output integration: the attention output is combined with the original input features and forwarded to subsequent strategy evaluation or decision-making modules.

Through this weighted aggregation mechanism, self-attention flexibly integrates global information into each time step, effectively enhancing the ability of the model to assess both historical strategies and the current game situation.

## 5. EXPERIMENTS

### 5.1. Experimental setup

To evaluate model performance, we adopt a role-swapping match method, following the experimental design outlined in the DouZero paper. Specifically, algorithm A and algorithm B alternately assume the roles of the landlord and peasants using a fixed set of hands. After completing a match, the roles are reversed – A switches to the peasant role while B becomes the landlord – while maintaining the same set of hands. This method, as validated in the DouZero study, effectively mitigates the influence of randomness, enhances experimental comparability, and facilitates a more precise assessment of the performance of different algorithms across roles.

Each test comprises 10 000 matches, with each match consisting of two rounds (one for each role swap) to ensure fairness and robustness of the experimental results. To maintain data diversity, hands for each match are randomly generated. The evaluation metrics follow those established in the DouZero paper, namely winning percentage (WP) and average difference in points (ADP), where WP quantifies the overall win-loss performance of a model, and ADP measures its scoring capability under various game conditions.

We employ DouZero as the baseline model and compare the performance of different attention mechanisms (CBAM, ECA, SE, SELF) within the DMC framework on both the peasant and landlord sides to analyze their impact on strategy optimization. The experimental results are summarized in Table 3, which illustrates WP and ADP trends across different modeling approaches. This systematic evaluation examines the influence of attention mechanisms on model performance and validates the effectiveness of our optimization methods.

All models in this study were implemented using PyTorch 2.0.0. The training process employed the Adam optimizer with an initial learning rate of 0.0001 and a batch size of 32. A gradient clipping threshold of 40 was applied to prevent gradient explosion and enhance training stability. During training, a fixed unroll length of 100 was used, and each parameter update was counted as one training step, which served as the primary metric for tracking training progress. Additionally, the exploration rate was set to 0.01 to balance exploration and exploitation in the strategy optimization process.

The experiments are executed on a Tesla A40 GPU, with CUDA version 11.8. The deep learning framework utilized is PyTorch 2.0.0, with numerical computations handled via NumPy 1.23. The whole experimental code is adapted from DouZero, incorporating attention mechanisms to enhance decision-making capabilities. Table 2 provides an overview of the model abbreviations used in the experiments – for example, DouZero-P denotes the original DouZero model, DouZero-T represents a retrained DouZero model, while X-F and X-Fmr indicate configurations where attention mechanisms are applied to different roles.

**Table 2**
Abbreviations and descriptions used in the experiment

| Abbreviation | Description |
|---|---|
| DouZero-P | Original DouZero model |
| DouZero-T | Retrained DouZero model |
| X-F | Both farmer and landlord use X attention |
| X-Fmr | Both farmer and landlord use X attention |
| A vs B | Model A plays as the landlord, Model B plays as the peasant |

### 5.2. Performance evaluation of attention mechanisms on the peasant side only

In this experiment, we investigate the effects of incorporating attention mechanisms exclusively on the peasant side while maintaining the landlord side with the original LSTM structure. The objective is to assess how attention mechanisms influence the peasant role and whether changes in the modeling approach of the peasant affect the performance of the landlord. The experimental results are depicted in Fig. 9 and Fig. 10, where Fig. 9 presents WP and ADP for the peasant, while Fig. 10 displays WP and ADP for the landlord to analyze any potential cross-role impact arising from peasant-side modifications.

As illustrated in Fig. 9a, applying CBAM solely to the peasant leads to a substantial increase in WP as compared to the original DouZero model. This suggests that CBAM enhances the strategic execution of the peasant, improving competitiveness and increasing the probability of winning. Figure 9b further demonstrates a rise in ADP, indicating that the peasant not only experiences an improvement in win rate but also achieves greater

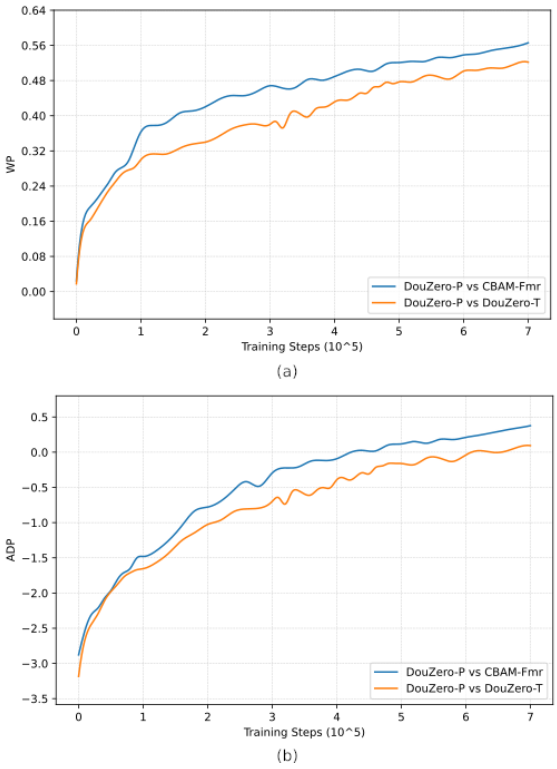


**Fig. 9.** Performance comparison of CBAM-Fmr and DouZero-T on the farmer role in terms of WP and ADP over training steps. (a) WP trend. (b) ADP trend

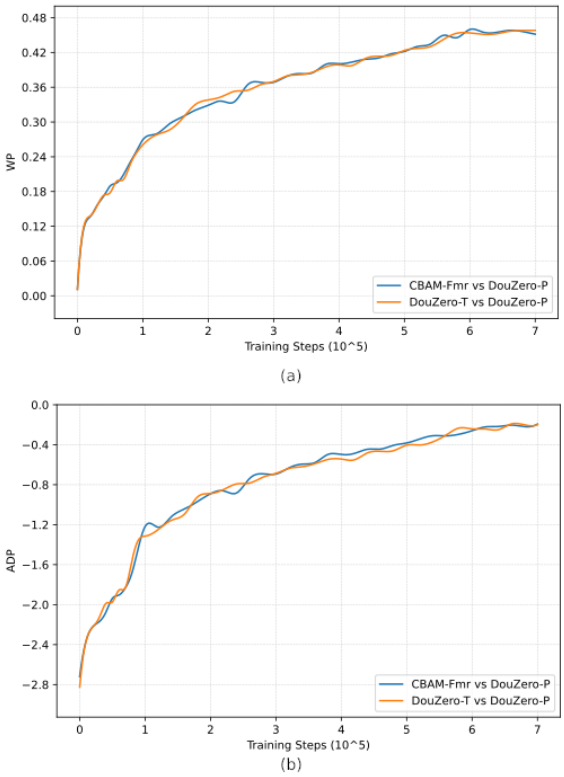


**Fig. 10.** Performance comparison of CBAM-Fmr and DouZero-T on the landlord role in terms of WP and ADP over training steps. (a) WP trend. (b) ADP trend

consistency in overall performance. These findings suggest that integrating CBAM refines the decision-making process of the peasant, facilitating more effective strategy execution and yielding higher rewards.

Figure 10a demonstrates that introducing CBAM on the peasant side does not result in a noticeable change in the WP of the landlord. This indicates that optimizations on the peasant side exert minimal influence on the overall gameplay performance of the landlord. Figure 10b further substantiates this observation, as the ADP difference remains marginal, confirming that the decision-making process of the landlord is largely unaffected by modifications to the peasant model. These results suggest that within the DMC framework, the strategic execution mechanisms of landlords and peasants exhibit a considerable degree of independence.

Based on these findings, we conclude that applying CBAM only to the peasant side significantly improves the win rate and scoring capability of the peasant, demonstrating that attention mechanisms enhance decision-making and lead to higher rewards. Meanwhile, the peasant-side optimizations exert negligible impact on the performance of the landlord, as the win rate and scoring metrics of the landlord remain largely unchanged. This implies that the modeling approach of the peasant can be refined independently without necessitating concurrent modifications to the strategy model of the landlord.

These findings highlight the feasibility of independently optimizing the model of the peasant while necessitating further exploration to evaluate the applicability of such modifications for the role of the landlord.

### 5.3. Performance analysis with attention on both roles

In 5.2, we examined the impact of introducing CBAM only on the peasant side, and observed an improvement in WP and ADP of the peasant, while the performance of the landlord remained largely unaffected. This suggests that the modeling approach of the peasant can be optimized independently without requiring simultaneous modifications to the strategy model of the landlord. However, within the DMC framework, it remains uncertain whether the landlord role can also benefit from attention mechanisms. If CBAM is applied to the landlord as well, will it achieve similar improvements as those seen on the peasant side, or will it negatively affect overall gameplay? Furthermore, will changes in the modeling approach of the landlord influence the decision-making ability of the peasant? To further explore these questions, we compared CBAM-F, CBAM-Fmr and DouZero-P to analyze the impact of landlord-side modeling on overall gameplay. The experimental results are shown in Fig. 11 and Fig. 12.

As seen in Fig. 11a, when CBAM-F plays as the peasant, its WP is significantly higher than DouZero-P and nearly identical to CBAM-Fmr. This suggests that changes in the modeling approach of the landlord do not significantly affect the strategic execution of the peasant. Similarly, Fig. 11b shows that CBAM-F achieves a higher ADP than DouZero-P, with only a slight difference from CBAM-Fmr, further confirming that modifica-

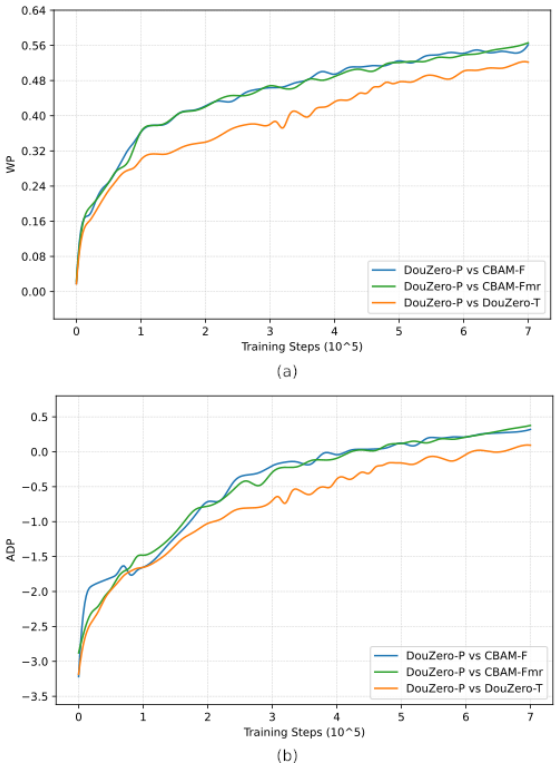


**Fig. 11.** Comparison of CBAM-F and CBAM-Fmr with DouZero-T on the farmer role. (a) WP performance over training steps. (b) ADP performance over training steps

tions on the landlord side have a minimal impact on the peasant. These findings align with the conclusion from Section 5.2 that the model of the peasant can be optimized independently without requiring simultaneous adjustments to the model of the landlord.

However, Fig. 12a reveals that when CBAM-F plays as the landlord, its WP is lower than both CBAM-Fmr and DouZero-P, suggesting that fully replacing LSTM with CBAM may weaken the decision-making ability of the landlord. This trend is further confirmed by Fig. 12b, where the ADP of CBAM-F is lower than both CBAM-Fmr and DouZero-P, indicating a decrease in overall reward levels. These results suggest that CBAM may not effectively enhance performance on the landlord side and may even lead to strategy degradation.

The observed decline in landlord performance could stem from fundamental differences in information modeling between LSTM and CBAM. Landlord decision-making typically involves long-term dependencies, where the strong capability of LSTM in modeling sequential information allows it to effectively capture the long-term influence of historical states. In contrast, CBAM, as an attention-based mechanism, primarily enhances local feature representations but may struggle to model the global strategy required for the landlord role, leading to weakened strategic execution. While the peasant role primarily relies on local information for decision-making, the landlord may depend more on long-term strategy planning. Thus, completely replacing LSTM on the landlord side may impair its ability to make long-term strategic decisions.

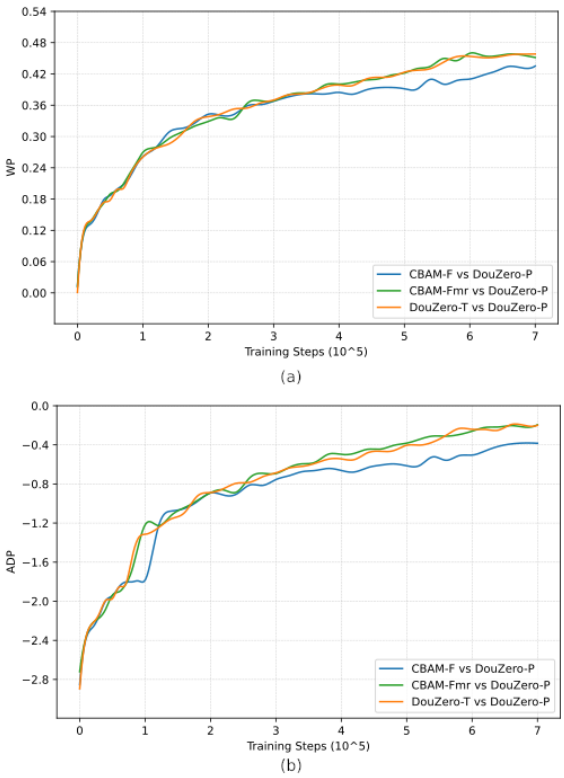


**Fig. 12.** Comparison of CBAM-F and CBAM-Fmr with DouZero-T on the landlord role. (a) WP performance over training steps. (b) ADP performance over training steps

### 5.4. Comparative experiments and analysis of different attention mechanisms

In 5.2 and 5.3, we observed that when distinct modeling approaches were applied to the peasant and landlord roles, their mutual influence was relatively limited. Furthermore, the experiments in Section 4.3 demonstrated that the CBAM mechanism improved performance on the peasant side but led to a decline on the landlord side. Therefore, this section further investigates the effects of different attention mechanisms in the DMC framework on both the peasant and landlord roles to comprehensively assess the applicability of attention mechanisms within the DMC structure. Experimental results are presented in Table 3. All results are reported with 95% confidence intervals computed over the evaluation matches to provide statistical reliability for the comparisons.

On the peasant side, introducing attention mechanisms generally enhances strategic execution, though the degree of improvement varies among mechanisms. CBAM yields the most pronounced enhancement, with WP increasing by 4.78% and ADP improving by 0.283, indicating that CBAM substantially strengthens the decision-making capability of the peasant, making it more competitive. ECA and SE also contribute to performance gains, with WP rising by 1.78% and 2.31% and ADP increasing by 0.11 and 0.137, respectively, though their improvements are less pronounced as compared to CBAM. Self-attention exhibits the weakest performance, with WP and ADP showing negligible differences from DouZero, suggesting that this mechanism provides limited benefits to the peasant role.

These findings indicate that CBAM, ECA and SE enhance strategic execution on the peasant side, whereas self-attention is relatively ineffective in the DouDizhu task. However, the influence of attention mechanisms on the landlord side follows a different trend. Incorporating attention mechanisms generally results in performance degradation, reinforcing the notion that LSTM remains the more suitable modeling approach for this role. Specifically:

CBAM, despite its strong performance on the peasant side, leads to a 2.31% reduction in WP and a 0.185 decline in ADP on the landlord side, suggesting its ineffectiveness in optimizing the strategy of the landlord. ECA and SE exert a more pronounced negative impact on the landlord, with WP decreasing by 3.25% and 2.95% and ADP dropping by 0.198 and 0.163, respectively, indicating a notable performance decline. Self-attention performs the worst on the landlord side, with WP decreasing by 3.74% and ADP dropping by 0.29.

One potential explanation for this phenomenon lies in the differences between the strategic characteristics of the landlord and peasant roles. The landlord acts as a single player competing against two coordinated opponents and must continuously track the evolution of the game state in order to infer potential strategies of the other players. Therefore, the landlord's decision-making process relies heavily on modeling the sequential dynamics of the game, especially the patterns formed by recent actions and opponent responses. Recurrent architectures such as LSTM are particularly suitable for this type of sequential reasoning because they explicitly preserve temporal

**Table 3**
WP and ADP performance of different methods against DouZero-P for farmers and landlords

| | Farmer WP (95% CI) | Farmer ADP (95% CI) | Landlord WP (95% CI) | Landlord ADP (95% CI) |
|---|---|---|---|---|
| DouZero-T | 51.81% [48.85%, 52.76%] | +0.094 [+0.077, +0.107] | 45.82% [43.87%, 47.78%] | −0.199 [−0.215, −0.185] |
| CBAM | 56.59% [54.48%, 57.66%] | +0.377 [+0.360, +0.391] | 43.51% [42.06%, 45.33%] | −0.384 [−0.401, −0.376] |
| ECA | 53.59% [51.65%, 55.51%] | +0.204 [+0.187, +0.215] | 42.57% [40.76%, 43.77%] | −0.397 [−0.415, −0.392] |
| SE | 54.12% [52.65%, 55.49%] | +0.231 [+0.215, +0.245] | 42.87% [41.05%, 44.68%] | −0.362 [−0.381, −0.349] |
| SELF | 51.23% [49.28%, 53.16%] | +0.092 [+0.074, +0.106] | 42.08% [40.27%, 43.87%] | −0.489 [−0.505, −0.476] |

dependencies in the decision process. In contrast, the attention modules investigated in this work, including CBAM, SE and ECA, mainly function as feature reweighting mechanisms that emphasize important channels or spatial components of the input representation. While such mechanisms can enhance the representation of important features, they do not explicitly model the temporal evolution of game states. Consequently, introducing these modules may partially weaken the sequential modeling capability required by the landlord role, which can reduce the stability of strategic decision-making.

Regarding the self-attention mechanism, previous studies have demonstrated that self-attention is highly effective at capturing long-range global dependencies within input representations. However, decision-making in DouDizhu does not primarily depend on long-range historical information across the entire game. Instead, it is largely influenced by local sequential context, particularly the most recent moves played by each participant. In such scenarios, architectures designed to model sequential dynamics, such as LSTM, may provide a more stable representation of the evolving game state than global dependency modeling mechanisms.

In addition to performance comparisons, we further analyze the computational cost of different model architectures to evaluate their practical applicability. Table 4 reports the parameter size and average inference time of each model. The LSTM architecture corresponds to the original backbone used in DouZero, while the other models incorporate additional attention mechanisms for comparison. As shown in Table 4, introducing attention mechanisms leads to moderate variations in computational complexity. CBAM exhibits the largest parameter size (1.66 M) and slightly higher inference latency (1.794 ms) due to the combined channel and spatial attention operations. In contrast, SE and ECA are relatively lightweight, maintaining smaller parameter sizes and faster inference speeds. Self-attention also introduces additional computation but remains comparable to the baseline model.

**Table 4**
Computational cost of different attention mechanisms

| | Params (M) | Inference time (ms) |
|---|---|---|
| LSTM | 1.46 | 1.745 |
| CBAM | 1.66 | 1.794 |
| ECA | 1.33 | 1.35 |
| SE | 1.48 | 1.227 |
| SELF | 1.37 | 1.488 |

Despite the modest increase in computational overhead, CBAM achieves the most significant performance improvements on the peasant side, as discussed earlier in this section. Considering both performance gains and computational cost, CBAM provides a favorable trade-off between decision quality and efficiency within the DMC framework.

### 5.5. Experiments and performance analysis against existing methods

In 5.2 and 5.3, we explored the performance impact of introducing attention mechanisms exclusively on the peasant side and on both roles simultaneously, respectively. The results demonstrated distinct strategic modeling patterns between peasants and landlords. While attention mechanisms provided significant optimization benefits for peasants, they actually impaired performance when applied to the landlord role. Based on these findings, and in conjunction with the comparative analysis of different attention mechanisms in 5.4, the DouRD model ultimately adopts CBAM on the peasant side while retaining the original DouZero structure on the landlord side. This configuration is designed to maximize the optimization benefits for peasants while preserving the stability and long-term dependency modeling capabilities required by the landlord.

To further verify the practical effectiveness of the proposed DouRD method and assess its relative advantages among current state-of-the-art DouDizhu agents, we designed two additional sets of experiments, with results summarized in Table 5 and Table 6. All results are reported with 95% confidence intervals computed over the evaluation matches to provide statistical reliability for the comparisons.

Table 5 centers on DouRD, presenting its head-to-head performance against other methods to evaluate its competitive strength in actual gameplay scenarios. Table 6 uses DouZero as a baseline and compares the performance changes of each method relative to it, thereby characterizing the extent of improvements achieved by DouRD over existing approaches.

This dual-perspective evaluation enables a comprehensive assessment of the competitiveness of DouRD and facilitates positioning the contribution of DouRD within the broader methodological landscape. Such a twofold analysis enhances inter-

**Table 5**
Performance of methods relative to DouRD

| | Farmer WP (95% CI) | Farmer ADP (95% CI) | Landlord WP (95% CI) | Landlord ADP (95% CI) |
|---|---|---|---|---|
| DouRD vs Douzero | 56.59% [54.48%, 57.66%] | +0.377 [+0.360, +0.392] | 45.16% [43.28%, 46.71%] | −0.193 [−0.211, −0.181] |
| DouRD vs Douzero | 54.02% [51.97%, 55.03%] | +0.246 [+0.228, +0.251] | 47.02% [45.12%, 48.88%] | −0.160 [−0.178, −0.148] |
| DouRD vs DouRN | 54.23% [52.18%, 55.71%] | +0.254 [+0.236, +0.263] | 47.52% [45.61%, 48.85%] | −0.132 [−0.150, −0.119] |
| DouRD vs SL | 78.12% [76.58%, 79.66%] | +1.512 [+1.495, +1.522] | 56.49% [54.39%, 58.54%] | +0.304 [+0.287, +0.311] |
| DouRD vs CQN | 88.43% [86.13%, 90.25%] | +2.403 [+2.386, +2.414] | 78.82% [76.32%, 81.27%] | +1.356 [+1.339, +1.363] |
| DouRD vs RHCP-v2 | 85.51% [83.32%, 87.15%] | +2.178 [+2.161, +2.194] | 68.24% [66.43%, 69.71%] | +1.512 [+1.495, +1.522] |
| DouRD Vs DeltaDou | 71.79% [70.11%, 73.68%] | +1.212 [+1.194, +1.223] | 49.14% [47.20%, 51.04%] | −0.354 [−0.371, −0.342] |

**Table 6**
Performance of methods relative to DouZero

| | Farmer WP (95% CI) | Farmer ADP (95% CI) | Landlord WP (95% CI) | Landlord ADP (95% CI) |
|---|---|---|---|---|
| Douzero vs Douzero | 52.19% [50.19%, 53.17%] | +0.094 [+0.077, +0.107] | 45.82% [43.87%, 47.78%] | −0.199 [−0.215, −0.185] |
| Douzero+ vs Douzero | 52.85% [50.86%, 54.11%] | +0.192 [+0.174, +0.204] | 46.85% [44.95%, 47.73%] | −0.123 [−0.139, −0.111] |
| DouRN vs Douzero | 52.46% [50.48%, 54.42%] | +0.208 [+0.191, +0.222] | 52.68% [50.66%, 54.37%] | +0.124 [+0.108, +0.137] |
| SL vs Douzero | 25.05% [23.57%, 26.52%] | −1.027 [−1.044, −1.014] | 43.71% [41.88%, 45.53%] | −0.294 [−0.311, −0.283] |
| CQN vs Douzero | 14.94% [13.65%, 17.22%] | −2.013 [−2.030, −2.001] | 21.29% [19.88%, 23.69%] | −1.352 [−1.369, −1.339] |
| RHCP-v2 vs Douzero | 16.88% [14.56%, 18.40%] | −1.901 [−1.918, −1.888] | 31.31% [29.71%, 32.90%] | −1.519 [−1.535, −1.507] |
| DeltaDou vs Douzero | 31.47% [29.86%, 33.07%] | −0.872 [−0.889, −0.859] | 50.79% [49.51%, 52.44%] | +0.359 [+0.342, +0.371] |

pretability of the results and reinforces the practical significance of the proposed method.

As shown in Table 5, DouRD demonstrates a clear advantage over all compared methods on the peasant side, exhibiting superior performance in WP and ADP. This further validates the effectiveness of attention mechanisms in enhancing strategic modeling for the peasant role. Meanwhile, performance on the landlord side remains stable, with relatively minor gaps as compared to most other methods, suggesting that the proposed architecture has no substantial negative impact on how the landlord makes decisions.

Table 6 provides a complementary analysis from a reverse perspective, using DouZero as the reference baseline to illustrate the relative performance changes of each method. The results show that traditional approaches such as SL, CQN, and RHCP-v2 are substantially inferior to DouZero across multiple evaluation metrics. Although enhanced variants like DouZero+ and DouRN exhibit marginal improvements in certain metrics, their overall gains are limited and often inconsistent – particularly on the peasant side. In contrast, DouRD achieves significant improvements over DouZero in both the peasant and landlord roles, with particularly notable gains in the win rate and ADP for the peasant side. These results further affirm the overall superiority of DouRD within the DMC framework, and validate the effectiveness of its architectural design.

In summary, DouRD demonstrates strong generalization capability and practical performance, especially in optimizing strategies for the peasant role. Retaining the LSTM architecture on the landlord side ensures overall strategic stability, indicating that the proposed design effectively balances performance improvement and inter-role policy consistency. Compared to existing methods such as DouZero+ and DouRN, DouRD achieves more substantial and stable improvements in key metrics such as the win rate and ADP on the peasant side, highlighting its remarkable advantage in role-specific strategy modeling.

### 5.6. Feature importance analysis

To improve the interpretability of the proposed model, we further analyze the feature importance distribution of the agents. Figure 13 illustrates the feature importance visualization for both the landlord agent and the farmer agent. The horizontal axis represents the feature indices, which correspond to the state representation components defined in Table 1.

From the visualization results, it can be observed that both agents assign the highest importance to Feature 8, which corresponds to the recent move history in the game state representation. This indicates that historical move information plays a crucial role in the decision-making process. Since DouDizhu is a sequential and highly strategic card game, the most recent actions provide important contextual cues for predicting strategies of opponents and determining appropriate responses.

In addition to the shared emphasis on recent move history, differences in feature importance distribution can also be observed between the two roles. The landlord agent assigns relatively higher importance to Features 3 and 4, which correspond to the cards played by the two opponents. This suggests that the

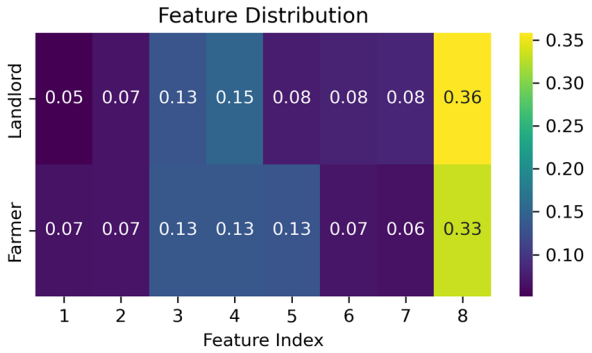


**Fig. 13.** Feature importance distribution

landlord tends to focus more on analyzing the recent actions of other players in order to infer their potential strategies and evaluate potential threats. Such behavior aligns with the landlord in the game, where the agent must carefully track opponent actions to maintain strategic advantage. In contrast, the farmer agent exhibits a relatively more balanced distribution of feature importance across several state components. Besides the played-card information, the farmer assigns comparable importance to multiple features in the state representation, indicating that its decision-making process relies on a broader range of contextual information. Such a strategy may help the farmer adapt more flexibly to different gameplay situations when coordinating against the landlord.

These observations provide empirical support for the mechanism analysis presented in Section 5.4. In particular, the strong emphasis on recent move history confirms that decision-making in DouDizhu relies heavily on short-term contextual information and sequential game dynamics, which further explains why sequential modeling architectures such as LSTM remain particularly effective for the landlord role.

### 5.7. Experimental summary

The above experiments demonstrate that attention mechanisms offer significant advantages in strategic modeling on the peasant side, with CBAM achieving the best performance in both WP and ADP, outperforming other mechanisms such as SE, ECA and self-attention. In contrast, applying attention mechanisms to the landlord side yields suboptimal results, with a general trend of performance degradation. This suggests that retaining the LSTM architecture remains more suitable for modeling sequential features on the landlord side.

Moreover, the results indicate a strong degree of independence between the modeling approaches for peasants and landlords, with minimal mutual influence observed during the respective strategy optimization processes. This characteristic implies that within the DMC framework, it is feasible to design optimal models tailored to each role without requiring unified structural adjustments, thereby improving overall training efficiency and deployment flexibility.

In comparative experiments with mainstream DouDizhu agents, the proposed DouRD method demonstrates superior performance on the peasant side across various benchmarks, while maintaining stable performance on the landlord side without evident degradation. The two-sided evaluation confirms not only the absolute performance improvements of DouRD but also its relative advantages within the current research landscape.

In conclusion, by integrating the CBAM attention mechanism on the peasant side and retaining the LSTM structure for the landlord side, DouRD effectively balances performance and stability. These findings validate the effectiveness of role-decoupled modeling under the DMC framework and offer a practically valuable reference for future strategy design in DouDizhu AI.

## 6. CONCLUSIONS AND FUTURE WORK

This study investigates the application of the DMC framework in imperfect information games, with particular focus on the influence of attention mechanisms in modeling DouDizhu agents. Experimental results indicate that introducing attention mechanisms on the peasant side significantly enhances strategy execution, with CBAM exhibiting the most pronounced optimization effect. However, on the landlord side, replacing LSTM entirely with an attention mechanism leads to performance degradation, suggesting that LSTM remains the more effective modeling approach for this role. Additionally, the results further validate the independence of the modeling approaches for the peasant and landlord, as optimizing the strategy model for one role does not substantially affect the performance of the other. This finding offers new insights into strategy optimization within the DMC framework.

While this study confirms the effectiveness of attention mechanisms for the peasant side, their applicability to the landlord side remains an open question. Future research could explore the following directions: (1) integrating LSTM with attention mechanisms to develop a more robust hybrid model for the landlord, balancing long-term planning capability with local decision-making efficiency; (2) evaluating different types of attention mechanisms (such as agent attention [27]) to assess their suitability for both the landlord and peasant roles, thereby identifying the most effective structure; (3) extending experiments to additional imperfect information game environments to verify the generalizability of the proposed approach.

## REFERENCES

[1] D. Silver *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018, doi: 10.1126/science.aar6404.

[2] D. Silver, *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016, doi: 10.1038/nature16961.

[3] C.B. Browne *et al.*, "A survey of monte carlo tree search methods," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, 2012, doi: 10.1109/TCIAIG.2012.2186810.

[4] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017, doi: 10.1038/nature24270.

[5] M. Moravcík *et al.*, "DeepStack: Expert-level artificial intelligence in heads-up no-limit poker," *Science*, vol. 356, no. 6337, pp. 508–513, 2017, doi: 10.1126/science.aam6960.

[6] N. Brown and T. Sandholm, "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals," *Science*, vol. 359, no. 6374, pp. 418–424, 2017, doi: 10.1126/science.aao1733.

[7] N. Brown and T. Sandholm, "Superhuman AI for multiplayer poker," *Science*, vol. 365, no. 6456, pp. 885–890, 2019, doi: 10.1126/science.aay2400.

[8] G. Lample and D.S. Chaplot, "Playing FPS games with deep reinforcement learning," in *Proc. 31st AAAI Conference on Artificial Intelligence*, 2017, pp. 2140–2146, [Online] Available: https://dl.acm.org/doi/10.5555/3298483.3298548 .

[9] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, doi: 10.1038/nature14236.

[10] K. Arulkumaran, A. Cully, and J. Togelius, "Alphastar: An evolutionary computation perspective," in *Proc. Genetic and Evolutionary Computation Conference Companion*, 2019, pp. 314–315, doi: 10.1145/3319619.3321894.

[11] C. Berner *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv, preprint arXiv: 1912.06680*, 2019, doi: 10.48550/arXiv.1912.06680.

[12] D. Zha *et al.*, "Rlcard: A toolkit for reinforcement learning in card games," in *Proc. Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2020, pp. 5264–5266, doi: 10.24963/ijcai.2020/764.

[13] V. Mnih, *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd International Conference on Machine Learning*, 2016, pp. 1928–1937, [Online]. Available: https://proceedings.mlr.press/v48/mniha16.html.

[14] Y. You, L. Li, B. Guo, W. Wang, and C. Lu, "Combinatorial q-learning for Dou Di Zhu," in *Proc. 16th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2020, pp. 301–307, doi: 10.1609/aiide.v16i1.7445.

[15] D. Zha *et al.*, "Douzero: Mastering doudizhu with self-play deep reinforcement learning," in *Proc. 38th International Conference on Machine Learning*, 2021, pp. 12333–12344, [Online]. Available: https://proceedings.mlr.press/v139/zha21a.html.

[16] N. Brown and T. Sandholm, "Solving imperfect-information games via discounted regret minimization," in *Proc. 33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 1829–1836, doi: 10.1609/aaai.v33i01.33011829.

[17] Q. Jiang, K. Li, B. Du, H. Chen, and H. Fang, "DeltaDou: Expert-level Doudizhu AI through Self-play," in *Proc. 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 1265–1271, doi: 10.5555/3367032.3367212

[18] Y. Zhao, J. Zhao, X. Hu, W. Zhou, and H. Li, "Douzero+: Improving doudizhu ai by opponent modeling and coach-guided learning," in *2022 IEEE Conference on Games*, 2022, pp. 127–134, doi: 10.1109/CoG51982.2022.9893710.

[19] Y. Chen, Y. Lyu, and D. Zhang, "DouRN: Improving DouZero by residual neural networks," in *2023 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2023, pp. 96–99, doi: 10.1109/CyberC58899.2023.00026.

[20] Q. Luo and T. Tan, "Improved learning efficiency of deep Monte-Carlo for complex imperfect-information card games," *Appl. Soft Comput.*, vol. 158, p. 111545, 2024, doi: 10.1016/j.asoc.2024.111545.

[21] S. Woo, J. Park, J.-Y. Lee, and I.S. Kweon, "CBAM: Convolutional block attention module," in *Computer Vision – ECCV 2018, Lecture Notes in Computer Science*, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[22] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[23] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11534–11542, doi: 10.1109/CVPR42600.2020.01155.

[24] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010, doi: 10.5555/3295222.3295349.

[25] D. Yang, W. Yang, M. Li, and Q. Yang, "Role-based attention in deep reinforcement learning for games," *Comput. Anim. Virtual Worlds*, vol. 32, no. 2, 2021, doi: 10.1002/cav.1978.

[26] L. Liu, X. Zhang, Z. He, and J. Liu, "Training a popular Mahjong agent with CNN and self-attention," *Int. J. Comput. Sci. Math.*, vol. 19, no. 2, pp. 157–166, 2024, doi: 10.1504/IJCSM.2024.137266.

[27] D. Han *et al.*, "Agent attention: On the integration of softmax and linear attention," in *Computer Vision–ECCV 2024*, 2024, pp. 124–140, doi: 10.1007/978-3-031-72973-7_8.