Scientific Research Publishing

Journal of Computer and Communications, 2025, 13(7), 67-93
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

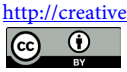
Multi-Agent Strategic Confrontation Game via Alternating Markov Decision Process Based Double Deep Q-Learning

Shou Feng, Chi Wei

Southwest China Institute of Electronic Technology, Chengdu, China
Email: shoufeng_cetc@126.com

How to cite this paper: Feng, S. and Wei, C. (2025) Multi-Agent Strategic Confrontation Game via Alternating Markov Decision Process Based Double Deep Q-Learning. *Journal of Computer and Communications*, **13**, 67-93.
https://doi.org/10.4236/jcc.2025.137004

Received: June 19, 2025
Accepted: July 14, 2025
Published: July 17, 2025



Abstract

To provide quantitative analysis of strategic confrontation game such as cross-border trades like tariff disputes and competitive scenarios like auction bidding, we propose an alternating Markov decision process (AMDP) based approach for modeling sequential decision-making behaviors in competitive multi-agent confrontation game systems. Different from the traditional Markov decision process typically applied to single-agent systems, the proposed AMDP approach effectively captures the sequential and interdependent decision-making dynamic characteristics of complicated multi-agent confrontation environments. To address the high-dimensional uncertainty resulting from the continuous decision-making space, we integrate the deep double Q-value network (DDQN) learning into the AMDP framework, leading to the proposed AMDP-DDQN approach. This integration enables agents to effectively learn their respective optimal strategies in an unsupervised manner to approximately solve the optimal policy problem, thereby enhancing decision-making quality in strategic confrontation tasks. As such, the proposed AMDP-DDQN method not only accurately predicts the confrontation game outcomes in sequential decision-making, but also provides dynamic and data-driven decision support that enables agents to effectively adjust their strategies in response to evolving adversarial conditions. Experimental results involving a strategic confrontation scenario between two countries with different situations of security, economy, technology, and administration demonstrate the effectiveness of the proposed approach.

Keywords

Deep Reinforcement Learning, Double Q-Value Network, Multi-Agent Strategic Confrontation Game, Alternating Markov Decision Process

1. Introduction

Strategic confrontation game [1] refers to competitive interactions among multiple agents, each aiming to maximize its own interests or objectives through adversarial decision-making and strategic actions. Typical examples include cross-border trades (e.g., tariff disputes), competitive market scenarios (e.g., auction bidding, price competition), and negotiation or bargaining processes, etc. Such competitive dynamics in confrontation game fundamentally differ from cooperative scenarios, as each participant in confrontation game system explicitly seeks superiority or dominance rather than mutual benefit in cooperative game. Among these contexts, international relations are a prominent example of strategic confrontation game and characterized by sequential and competitive interactions among nation-states across security, economy, technology, and administration [2]. Due to the high uncertainty and profound consequences inherent in strategic confrontation games, it is crucial to develop systematic models of such competitive interactions using rigorous quantitative methods. As such, effective modeling of strategic confrontation game not only enhances understanding of adversarial sequential behavior, but also offers strategically meaningful insights and reliable decision-making support for policymakers engaged in international strategic competition.

Game theory, developed in the early 20th century, has become a cornerstone for advancing the understanding of strategic interactions under competitive settings [3] [4]. The seminal work of Von Neumann and Morgenstern [5] has established a rigorous mathematical foundation for analyzing decision-making in conflict scenarios. Building upon this foundation, the development of extensive-form games (EFGs) [6] [7] significantly extends the descriptive capability of game theory by modeling multi-stage strategic interactions for sequential decision-making process through tree-structured representations. In addition, Bayesian game [8] addresses problems with incomplete or asymmetric information by introducing belief systems and probabilistic reasoning into strategic decision-making. This framework allows the agent to update its strategy based on private information and expectations about other agents, making it particularly useful for modeling uncertainty in strategic confrontation game. Another line of advanced methods for decision-making includes the evolutionary game theory [9] which shows how strategies evolve over time or potentially punish past behaviors through mechanisms such as replication, mutation, and selection. As a practical implementation of strategic modeling, the RAND Strategic Assessment System (RSAS) [10]-[12] represents an early effort to simulate strategic decision-making, notably between the United States and the Soviet Union during the Cold War. While RSAS illustrates the practical utility of game-theoretic simulation in national security analysis, its reliance on expert-defined rules and lack of learning mechanisms limit its ability to model the uncertain dynamic behavior of modern strategic confrontations.

Despite certain appealing theoretical features, classical game-theoretic models [5]-[9] are subject to inherent limitations when addressing modern strategic con-

frontation scenarios, particularly for those involving high uncertainty and multi-agent dynamics in confrontation international relations. In fact, real-world confrontations often feature high-dimensional state and action spaces, substantial uncertainty for observations, and non-equilibrium behaviors among multiple competitive agents. Classical game theory typically assumes rational decision-making and equilibrium-based assumptions that rarely hold in complicated competitive confrontation environments. Consequently, there is an increasing demand for flexible and scalable frameworks that can effectively model the sequential, adversarial, and uncertain characteristics for strategic confrontation game.

To overcome the limitations of classical game-theoretic approaches in modeling sequential and uncertain decision-making, Markov decision process (MDP), introduced by Bellman in the 1950s [13], provide a foundational framework for representing agent behavior under uncertainty through probabilistic state transitions and reward-driven optimization. Then, the development of partially observable Markov decision process (POMDP) [14] further extends this framework to the case with incomplete or noisy observations. Building on these foundations, the multi-agent reinforcement learning based approaches [15]-[17] have achieved promising results in adversarial decision-making for modeling multi-agent game systems such as in extended Boid modeling for drone or UAV swarms [18]-[20]. However, these methods typically rely on the assumption of shared goals or aligned incentives among agents. This fundamentally limits their applicability to adversarial domains like strategic confrontations such as in international relations, where agents usually pursue conflicting objectives and seek to maximize their advantage over opponents rather than cooperate toward mutual benefit.

In recent years, deep reinforcement learning (DRL) [21]-[24], as an in-depth combination of artificial neural network and reinforcement learning, has opened new avenues for modeling complicated, high-dimensional, sequential decision-making problems. By combining the representational power of deep learning for extracting high-level features with the adaptive decision-making capability of reinforcement learning, DRL enables agents to learn optimal policies directly from raw observations [16] [25]. This makes it particularly suitable for modeling dynamic environments in which both perception and strategy are critical, such as confrontation games, control tasks, and strategic planning scenarios. A key milestone in DRL is the deep Q-network (DQN) introduced by Mnih *et al.* in [26], which exploits deep neural networks to approximate value functions from raw data, thereby enabling agents to attain near-human performance in Atari games. Followed by DQN, the deep double Q-Network (DDQN) [27] is developed to mitigate the overestimation bias of Q-values in standard DQN. For problems involving continuous action spaces, deterministic policy learning becomes essential. To this end, the deep deterministic policy gradient method [28] leverages an actor-critic architecture with neural function approximations to learn deterministic policies for handling high-dimensional and continuous decision-making problems.

Despite significant progress, most existing DRL applications are formulated for

single-agent systems or cooperative multi-agent settings, where agents either pursue shared objectives or operate under simultaneous action assumptions. However, these formulations are not well-suited for adversarial environments—particularly for those characterized by alternating or turn-based decision-making, as commonly observed in strategic confrontation game between nation-states or rival agents. As a result, standard DRL frameworks often fall short in capturing the sequential dependencies, inter-agent strategy interplay, and explicit competitive dynamics inherent in such confrontational settings. The limitations of existing DRL methods call for specialized DRL frameworks that can effectively handle alternating and adversarial scenarios.

To enable rigorous quantitative modeling of strategic confrontation game, this paper develops an alternating Markov decision process (AMDP) based approach that is explicitly designed to model sequential and adversarial interactions among competitive agents. Unlike traditional MDP [13] and POMDP [14], the proposed AMDP framework inherently accounts for the turn-based decision structure in multi-agent strategic confrontation game. Furthermore, to address the high-dimensional uncertainty resulting from the continuous action spaces, we integrate the DDQN learning [27] into the AMDP framework (named as the AMDP-DDQN) to effectively learn their respective optimal strategies and enhance decision-making quality in strategic confrontation game. Although DDQN has been widely applied across various disciplines (see, e.g., [29]-[33]), previous works seldom concern its use within the multi-agent strategic confrontation game under the AMDP paradigm. Overall, the main contributions of this paper are summarized as follows:

1) We propose an AMDP based framework for modeling sequential strategic interactions among completive agents, which can effectively capture the sequential nature and interactive characteristics of multi-agent strategic confrontation game to enable adaptive and interdependent decision-making.

2) We integrate the DDQN based deep reinforcement learning within the AMDP framework to approximately maximize the intractable action value objective function and efficiently enable agents to learn optimal adversarial strategies for decision-making in strategic confrontation scenarios.

3) We conduct various numerical experiments to demonstrate the effectiveness (including crisis prediction and strategy evaluation) of the proposed approach in a confrontation game scenario between two nations with different situations of security, economy, technology, and administration.

2. AMDP Based Problem Modeling of Multi-Agent Strategic Confrontation Game

Basically, a multi-agent strategic confrontation game system is a dynamic or sequential interaction process where multiple players take turns acting under some specific conditions. In this system, all players can observe actions of previous players before choosing their own strategies, enabling adaptive and interdependent

decision-making. To mathematically model these interaction dynamics, we propose an alternating Markov decision process (AMDP) based approach, an extension of the conventional Markov decision process (MDP), which can effectively capture the sequential nature and interactive characteristics inherent to the multi-agent strategic confrontation game. The technical details of the AMDP formulation are elaborated as follows.

The proposed AMDP comprises N ordered players, namely Player 1, Player 2, …, Player N, where each player represents an agent with autonomous decision-making capability. That is to say, it operates through sequential and turn-based interactions: Starting from Player 1, only one player is allowed to act according to some designed regulations at each time step, and all players act alternately in order until the end of the game. Specifically, the AMDP is defined as a five-tuple $\langle N, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ represent the spaces of all possible states and actions of all N players, respectively. Meanwhile, we use $\boldsymbol{s}_t \in \mathcal{S}$ and $\boldsymbol{a}_t^i \in \mathcal{A}$ to stand for the state and the action adopted by Player i at time t, respectively. In addition, $\mathcal{P}$ and $\mathcal{R}$ denote the state transition function set and the reward function set, respectively. Furthermore, we use $p(\boldsymbol{s}' \mid \boldsymbol{s}, \boldsymbol{a}) \in \mathcal{P}$ to denote the conditional (state transition) probability to the state $\boldsymbol{s}'$ when adopting action $\boldsymbol{a}$ in state $\boldsymbol{s}$ and $r(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') \in \mathcal{R}$ to represent the reward obtained by taking action $\boldsymbol{a}$ in the state $\boldsymbol{s}$ and the transition state $\boldsymbol{s}'$. Such a mathematical formulation provides a well-defined and structured foundation for facilitating subsequent learning and strategic decision-making in sequential and competitive multi-agent environments.

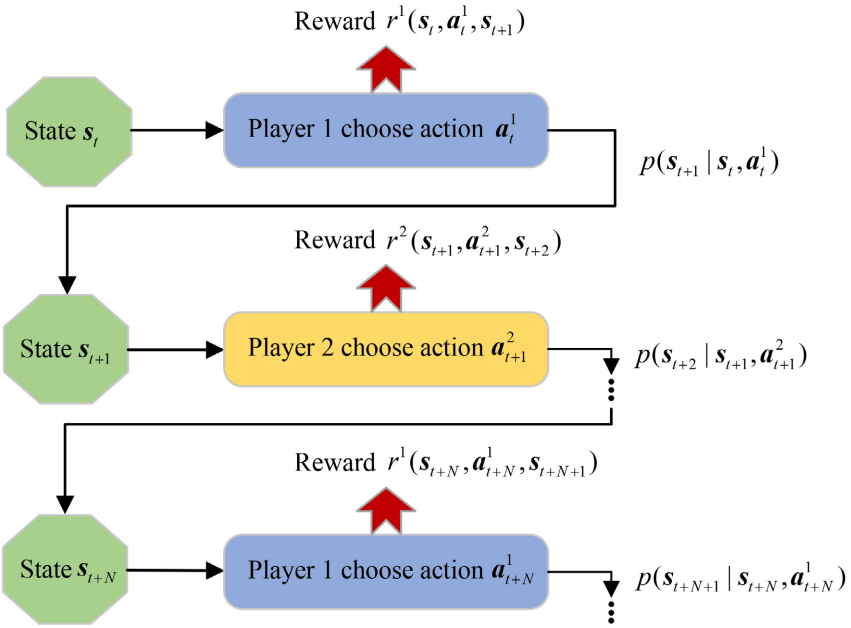


Figure 1. The proposed AMDP workflow.

The detailed AMDP workflow is illustrated in **Figure 1**. At time t, Player 1 observes the current state $\boldsymbol{s}_t$ and executes action $\boldsymbol{a}_t^1$, and then receives the cor-

responding reward $r\left(\boldsymbol{s}_t, \boldsymbol{a}_t^1, \boldsymbol{s}_{t+1}\right)$, where the transferred state $\boldsymbol{s}_{t+1}$ is obtained according to the state transfer function $p\left(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{a}_t^1\right)$. Next, the turn order proceeds to Player 2. Following the same procedure as that of Player 1, Player 2 observes the state $\boldsymbol{s}_{t+1}$, takes action $\boldsymbol{a}_{t+1}^2$, and receives the reward $r\left(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}^2, \boldsymbol{s}_{t+2}\right)$. Such a turn order proceeds sequentially through all players until Player N acts and take actions. Following the turn of Player N, the cycle restarts with Player 1 and repeats the same procedure iteratively until the game terminates or the termination condition is reached.

For Player n, we define its policy $\pi_n\left(\boldsymbol{s}\right)=\boldsymbol{a}$ as a function mapping from a state $\boldsymbol{s} \in \mathcal{S}$ to an action $\boldsymbol{a} \in \mathcal{A}$. The objective of Player n in AMDP based game is to obtain its optimal policy by maximizing the expectation (action-value function) of the following cumulative reward

$$G_t^n = \sum_{k=0,N,2N,\cdots} \gamma^{\frac{k}{N}} r\left(\boldsymbol{s}_{t+k}, \boldsymbol{a}_{t+k}^n, \boldsymbol{s}_{t+k+1}\right), \tag{1}$$

where $\gamma \in (0,1]$ denotes the discount factor to regularize the trade-off between the immediate and future rewards. Thus, the action-value function can be calculated as

$$\begin{aligned} Q^{\pi_n}\left(\boldsymbol{s},\boldsymbol{a}\right) &= \mathbb{E}\left[G_t \mid \boldsymbol{s}_t=\boldsymbol{s}, \boldsymbol{a}_t=\boldsymbol{a}, \pi_n\right] \\ &= r\left(\boldsymbol{s}_t, \boldsymbol{a}_t^n, \boldsymbol{s}_{t+1}\right) \\ &\quad + \sum_{k=N,2N,\cdots} \gamma^{\frac{k}{N}} r\left(\boldsymbol{s}_{t+k}, \boldsymbol{a}_{t+k}^n, \boldsymbol{s}_{t+k+1}\right) \prod_{j=0}^{k+N-1} p\left(\boldsymbol{s}_{t+j+1} \mid \boldsymbol{s}_{t+j}, \boldsymbol{a}_{t+j}^{(j-1) \bmod N}\right), \end{aligned} \tag{2}$$

where mod denotes the modulo operator that returns the remainder after dividing one number by another. Essentially, the action-value function $Q^{\pi_n}\left(\boldsymbol{s},\boldsymbol{a}\right)$ in Equation (2) represents the expected return when the n-th agent (player) takes action $\boldsymbol{a}$ at the state $\boldsymbol{s}$ and exploits the policy π_n.

Now, the optimal policy problem for Player n is to find an optimal action by maximizing $Q^{\pi_n}\left(\boldsymbol{s},\boldsymbol{a}\right)$ in Equation (2) with respect to π_n, *i.e.*,

$$\pi_n^* = \arg\max_{\pi_n} Q^{\pi_i}\left(\boldsymbol{s},\boldsymbol{a}\right). \tag{3}$$

We notice that directly solving the optimal policy problem (3) is computationally intractable because the calculation of the objective function $Q^{\pi_n}\left(\boldsymbol{s},\boldsymbol{a}\right)$ inherently involves combinatorial search process, where the number of possible state-action configurations grows exponentially with the dimension of the state or action space. To deal with this, we resort to a deep reinforcement learning-based method to find a tractable approximation or an estimate of $Q^{\pi_n}\left(\boldsymbol{s},\boldsymbol{a}\right)$. The discussion of how to approximate the action-value function $Q^{\pi_n}\left(\boldsymbol{s},\boldsymbol{a}\right)$ will be elaborated in the next Section. To facilitate the optimal strategy training, in the following subsections we first discuss the designs of the state space, action space, state transfer function, and the reward function.

2.1. State Space Design

Note that each player in AMDP takes its own actions based on the actions of other

players. To capture these interdependence of actions of all players, we define the state vector at time t as

$$\boldsymbol{s}_t = \left\{ s_t^{n,m} \right\}_{n=1,\cdots,N;m=1,\cdots,M} \bigcup \left\{ \tilde{\boldsymbol{a}}_t^{-n} \right\}, \tag{4}$$

where $s_t^{n,m}$ is the m-th state of the n-th player at time t, N is the number of players, and each player has M states. Meanwhile, $\tilde{\boldsymbol{a}}_t^{-n}$ denotes the most recent action taken by other players and is defined as

$$\tilde{\boldsymbol{a}}_t^{-n} = \underbrace{\left\{ \boldsymbol{a}_{t-N+1}^{n+1}, \cdots, \boldsymbol{a}_{t-n}^{N}, \boldsymbol{a}_{t-n+1}^{1}, \cdots, \boldsymbol{a}_{t-1}^{n-1} \right\}}_{N-1 \text{ actions}}, \tag{5}$$

where the superscript $-n$ means all players except Player n and $\tilde{\boldsymbol{a}}_t^{-n}$ has a total of $N-1$ actions. The specific composition of the state vector $\boldsymbol{s}_t$ in Equation (4) is illustrated in Figure 2 for intuitive representation.

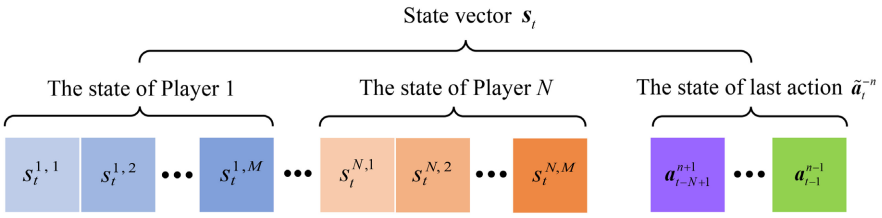


Figure 2. Composition of the state vector $\boldsymbol{s}_t$.

In addition, at the end of the strategic confrontation game we define the set of the game outcome as $\boldsymbol{c} = \{c_1, c_2, \cdots, c_W\}$, where W is the numbers of the possible outcome. Each outcome $c_w \left(w \in \{1, \cdots, W\} \right)$ of $\boldsymbol{c}$ is uniquely associated with a distinct set of terminal states $\boldsymbol{Z}_w \subset \mathcal{S}$, such that $\boldsymbol{Z}_{w_1} \cap \boldsymbol{Z}_{w_2} = \varnothing$ for any $w_1 \neq w_2$ where w_1, $w_2 \in \{1, 2, \cdots, W\}$. In other words, the game concludes with outcome c_w if and only if the current state $\boldsymbol{s}$ satisfies $\boldsymbol{s} \in \boldsymbol{Z}_w$, ensuring mutual exclusivity between outcomes.

2.2. Action Space Design

Directly modeling high-dimensional continuous action spaces for multi-agent decision-making often incurs prohibitive computational costs and challenges in model convergence. To address this, we adopt a composite two-dimensional action space, which decomposes complicated actions into hierarchical decisions. Specifically, the action space is designed as a composite two-dimensional space

$$\boldsymbol{a}_t^n = \left(x_t^n, y_t^n \right), \tag{6}$$

where $x_t^n \in \{1, \cdots, U\}$ and $y_t^n \in \{1, \cdots, L\}$ represent the action type and action degree selected by Player i at time t, K and L are the number of action types and action degrees, respectively. We notice that the action-space design in Equation (6) not only captures a diverse range of action types, but also quantifies their intensity, enabling the AMDP model to flexibly adapt to different levels of decision-making.

2.3. State Transfer Function Design

The state transfer function $p(\boldsymbol{s}' \mid \boldsymbol{s}, \boldsymbol{a})$ is essentially a conditional probability of the state $\boldsymbol{s}'$ given $\boldsymbol{s}$ and $\boldsymbol{a}$, and is designed as the following normal distribution

$$p(\boldsymbol{s}' \mid \boldsymbol{s}, \boldsymbol{a}) = \mathcal{N}\left(\mathcal{F}(\boldsymbol{s}, \boldsymbol{a}), \sigma^2\right), \tag{7}$$

where $\mathcal{F}(\boldsymbol{s}, \boldsymbol{a})$ and σ^2 denote its mean and variance, respectively. To characterize the interdependencies between actions, the mean $\mathcal{F}(\boldsymbol{s}, \boldsymbol{a})$ in Equation (7) is designed as

$$\mathcal{F}(\boldsymbol{s}, \boldsymbol{a}) = \mathcal{F}\left(s_t^{n,m}, \tilde{\boldsymbol{a}}_t^{-n}, \boldsymbol{a}\right) = f_{\left(\boldsymbol{a}_t^{-n}, \boldsymbol{a}\right)}\left(s_t^{n,m}\right), \tag{8}$$

where $f_{\left(\boldsymbol{a}_t^{-n}, \boldsymbol{a}\right)}$ represents the state transition function determined by the $\left(\boldsymbol{a}_t^{-n}, \boldsymbol{a}\right)$ tuple and can be set according to the actual physical meaning of the action in a specific scenario of strategic confrontation game.

2.4. Reward Function Design

In reinforcement learning, the reward function has an important role in guiding the sequential behavior of the agent, as it is usually regarded as the optimization criterion and ultimately shapes the learned policy for design-making. In this paper, the reward function for Player n, denoted as $r^n\left(\boldsymbol{s}_t, \boldsymbol{a}_t^n, \boldsymbol{s}_{t+1}\right)$, consists of the following two components:

$$r^n\left(\boldsymbol{s}_t, \boldsymbol{a}_t^n, \boldsymbol{s}_{t+1}\right) = r_1^n\left(\boldsymbol{s}_{t+1}\right) + r_2^n\left(\boldsymbol{s}_t, \boldsymbol{s}_{t+1}\right). \tag{9}$$

Here, the first term $r_1^n\left(\boldsymbol{s}_{t+1}\right)$ is a terminal reward and defined as

$$r_1^n\left(\boldsymbol{s}_{t+1}\right) = R_w^n, \text{ if } \boldsymbol{s}_{t+1} \in \boldsymbol{Z}_w, \tag{10}$$

where R_w^n stands for the obtained reward value of Player n when the state $\boldsymbol{s}_{t+1}$ at time $t+1$ belongs to the w-th terminate state set $\boldsymbol{Z}_w$. The second term $r_2^n\left(\boldsymbol{s}_t, \boldsymbol{s}_{t+1}\right)$ in Equation (9) reflects the state change-based reward, which encourages the progress of Player n while penalizing the progress of other players and is defined as

$$r_2^n\left(\boldsymbol{s}_t, \boldsymbol{s}_{t+1}\right) = \eta_1 \sum_{m=1}^{M}\left(\boldsymbol{s}_{t+1}^{n,m} - \boldsymbol{s}_t^{n,m}\right) - \eta_2 \sum_{\substack{j=1 \\ j \neq n}}^{N} \sum_{m=1}^{M}\left(\boldsymbol{s}_{t+1}^{j,m} - \boldsymbol{s}_t^{j,m}\right), \tag{11}$$

where N is the number of players, and M is the number of state dimensions for each player, and η_1 and η_2 are the weighting factors of own-state gain and opponent-state loss. The reward design in Equation (9) encourages each player to maximize its own reward while suppressing the advancement of its opponents, thereby promoting competitive strategic behavior.

It is worth noting that the terminal reward r_1 in Equation (10) is usually assigned with a positive value to indicate a favorable outcome and a negative value to indicate an unfavorable one for the associated agent. This aligns with the long-term confrontation strategy of the associated agent with a hope of game success. The stepwise reward r_2 in Equation (11) provides additional guidance during

the learning process by evaluating the quality of each state transition. In general, a positive reward is given if the post-transition state s' improves upon the previous transition state s. Such a reward design indeed encourages steady progress toward advantageous states while preserving the alignment with overarching strategic goals.

3. Agent Training via Double Deep Q-Learning

Reinforcement learning seeks the optimal strategy π^* [1] by interacting with the environment under the AMDP framework, aiming to approximately maximize the action value function $Q^{\pi_n}(s,a)$ defined in Equation (2) to guide the agent to make optimal decisions. The deep double Q-value network (DDQN) [27] is an improved reinforcement learning algorithm to suppress the overestimation problem in the original deep Q-Network. Overall, DDQN exploits two separated neural networks to effectively estimate Q-values independently.

In the proposed AMDP framework for multi-agent strategic confrontation game, each player is treated as an agent to learn an approximated $Q^{\pi_n}(s,a)$ with the input state s. Armed with the AMDP modeling introduced in the previous section, the proposed architecture of DDQN is illustrated in **Figure 3**, where each player contains two neural networks with a similar structure, namely the online network Q_1 and the target network Q_2, along with an experience replay pool D where we use $|D|$ to denote the number of samples in D. The agent alternatively interacts with the environment—which may include other agents using random strategy—to learn an approximated action value function of $Q^{\pi_n}(s,a)$ in Equation (2) and then find the optimal policy via solving problem (3). The detailed implementation of DDQN will be discussed in the subsequent subsections.

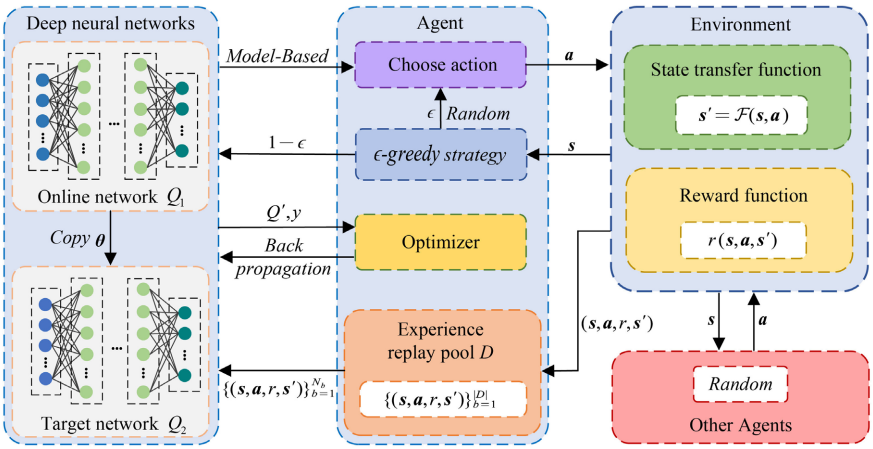


Figure 3. Architecture of DDQN.

3.1. Structure of Double Q-Value Neural Network

Based on the AMDP model introduced in Section 2, we implement a feed-forward

[1]For convenience, the subscript n is omitted here to indicate that the notation is applicable to any agent.

neural network architecture parameterized by an appropriate set of parameters $\boldsymbol{\theta}$, which adopts the state vector $\boldsymbol{s}_t$ at time t as input and obtains multiple estimated action-values $\{Q'(\boldsymbol{s}_t, \boldsymbol{a}; \boldsymbol{\theta})\}$. It should be noted that when learning a neural network, the dimensions of input and output must satisfy the dimensions of the state vector and the number of actions, respectively. The relationship between the input $\boldsymbol{s}_t$ and the output $Q'(\boldsymbol{s}_t, \boldsymbol{a}; \boldsymbol{\theta})$ within deep neural network of the double Q network is detailed in Figure 4. The parameter sets of Q_1 and Q_2 to be learned in the training phase are denoted by $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, respectively. Notice that the online network Q_1 and the target network Q_2 share the same neural network structure.

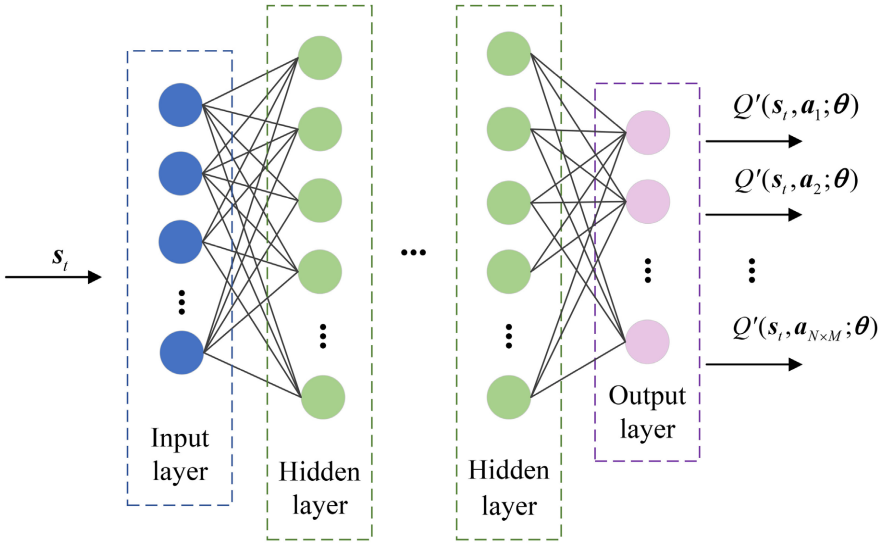


Figure 4. Input-output relationship of the double Q network.

3.2. Online Target Network Update

In reinforcement learning, unlike supervised learning, training data is not pre-collected but rather generated by agent through interactions with the environment and simultaneously is used to improve the agent's policy through continuous learning. Specifically, at the beginning of training, the corresponding agent observes the current state $\boldsymbol{s}$ and selects an action $\boldsymbol{a}$ that maximizes the Q-value estimated by the online network Q_1 to interact with the environment though the state transfer function $p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{a})$ and the reward function $r(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s})$. This interaction yields the next state $\boldsymbol{s}'$ and a corresponding reward value r. Each interaction result is preserved in the experience replay buffer D as a four-tuple sample $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}')$. Then, we take N_b (batch size) samples used for training when the number of samples in the replay buffer reaches the maximum sample capacity N_r.

In interactions between the agent and the environment, we adopt an epsilon-greedy strategy to ensure that the agent can explore the action space sufficiently to avoid suboptimal solutions caused by premature convergence. Specifically, the epsilon-greedy strategy based action selection is given by

$$\boldsymbol{a}=\begin{cases}\text{an action randomly selected from } \mathcal{A}, & \text{with probability } \epsilon,\\ \arg\max_{\boldsymbol{a}} Q'\left(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta}_1\right), & \text{with probability } 1-\epsilon,\end{cases} \tag{12}$$

where $\mathcal{A}$ represents the action space, $\boldsymbol{s}$ is the current state, and $Q'\left(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta}_1\right)$ is the estimated action-value function that is used to predict the expected cumulative reward when taking action $\boldsymbol{a}$ at state $\boldsymbol{s}$. The value of ϵ is decayed over time to allow for more exploration during the initial stages of training and gradually shift toward exploitation during the stable stage as the agent gains more confidence in its learned policy. Such a decay scheme in the λ-th episode is implemented as

$$\epsilon\left(\lambda\right)=\epsilon_{final}+\left(\epsilon_{start}-\epsilon_{final}\right)\mathrm{e}^{-\frac{\lambda}{\epsilon_{decay}}} \tag{13}$$

where ϵ_{start} and ϵ_{final} represent the value of $\epsilon\left(\lambda\right)$ at the initial and the stable stage of training, respectively, and ϵ_{decay} is used to control the decay rate of $\epsilon\left(\lambda\right)$.

Based on the epsilon-greedy strategy in Equation (12), the corresponding agent performs actions in response to environmental states, thereby generating a sequence of four-tuple samples that serve as the training data. For each four-tuple sample, the online network Q_1 is employed to estimate the Q-values $Q'\left(\boldsymbol{s}',\boldsymbol{a};\boldsymbol{\theta}_1\right)$ for all possible actions with the next state $\boldsymbol{s}'$ and select an action that maximizes $Q'\left(\boldsymbol{s}',\boldsymbol{a};\boldsymbol{\theta}_1\right)$ with respect to $\boldsymbol{a}$. The selected action in the online network Q_1 is then used in the target network Q_2 to compute the following the target Q-value

$$Q'\left(\boldsymbol{s}',\arg\max_{\boldsymbol{a}} Q'\left(\boldsymbol{s}',\boldsymbol{a};\boldsymbol{\theta}_1\right);\boldsymbol{\theta}_2\right). \tag{14}$$

Accordingly, we obtain the indirect estimate of the Q-value of the current state, which is given by the summation of the immediate reward r and the discounted target Q-value of the next state, that is,

$$y=r+\gamma Q'\left(\boldsymbol{s}',\arg\max_{\boldsymbol{a}} Q'\left(\boldsymbol{s}',\boldsymbol{a};\boldsymbol{\theta}_1\right);\boldsymbol{\theta}_2\right), \tag{15}$$

where $\gamma\in\left(0,1\right]$ denotes the discount factor.

So far, we have obtained the direct estimate of Q-value $Q'\left(\boldsymbol{s},a;\boldsymbol{\theta}_1\right)$ and a more accurate indirect estimate y of Q-value derived from the known reward r and the direct estimate of the next step. Assume that a batch of N_b samples are randomly drawn from the experience replay buffer D. Then, the loss function $\mathcal{L}$ of the neural network can be defined as the mean squared error between the two estimated Q values, that is,

$$\mathcal{L}\left(\boldsymbol{\theta}_1\right)=\mathbb{E}\left[\left(y-Q'\left(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta}_1\right)\right)^2\right]. \tag{16}$$

The training (online network update) of the neural network parameter aims to find the optimal parameter $\boldsymbol{\theta}_1^*$ by minimizing

$$\boldsymbol{\theta}_1^*=\arg\min_{\boldsymbol{\theta}_1}\mathcal{L}\left(\boldsymbol{\theta}_1\right). \tag{17}$$

It is worth noting that directly solving Problem (17) is generally intractable as the objective function $\mathcal{L}(\boldsymbol{\theta}_1)$ exhibits strong nonlinearity with respect to $\boldsymbol{\theta}_1$. Therefore, we adopt a random gradient-based scheme to iteratively approximate the solution. Specifically, we propose to employ the Adam optimizer [34], which has demonstrated strong performance for solving complicated nonlinear optimization inherent to deep reinforcement learning. To stabilize learning and suppress rapid fluctuations in the learning target, the parameter set $\boldsymbol{\theta}_1$ of the online Q-network, Q_1, is periodically synchronized with that of the target Q-network, Q_2, *i.e.*, $\boldsymbol{\theta}_2^* = \boldsymbol{\theta}_1^*$, within each N^- episodes.

3.3. The Overall Algorithm

Through sequentially interaction with the environment, the corresponding agent updates its policy to maximize the expected cumulative return. By decoupling the action selection from the target-value computation, the DDQN can effectively mitigate the Q-value overestimation bias, thereby enhancing training stability and boosting the ultimate policy decision-making performance. In conjunction with the AMDP formulation in Section 2, the entire DDQN based training procedure is outlined in **Algorithm 1**, which is hereafter referred to as the AMDP-DDQN. The computational complexity of AMDP-DDQN in **Algorithm 1** is dominated by $\mathcal{O}\left(N_{episode}N_{avg}N_bN_{state}\left(\sum_{i=1}^{N_H-1}H_iH_{i+1}\right)N_{action}\right)$, where N_{avg} denotes the number of averaging steps per episode, N_{state} represents the state-space dimensionality, N_{action} is the action-space cardinality, and H_i corresponds to the i-th hidden layer dimensions in DDQN, N_H indicates the number of hidden layers.

Algorithm 1. AMDP-DDQN.

Initialize the experience replay pool D with the maximum sample capacity N_r. Define the parameters of the online network Q_1 and the target network Q_2, which are initialized as $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, respectively, and we set $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$ in the initial training;
Set the batch size N_b, the target network update frequency N^-, the number of episode $N_{episode}$, and the maximum game step size N_s;
for $\lambda = 1, \ldots, N_{episode}$
  Initialize the state $\boldsymbol{s}$ and set $\tau = 0$;
  do
    $\tau \leftarrow \tau + 1$
    Randomly choose action $\boldsymbol{a}$ with probability ϵ; otherwise, $\boldsymbol{a} = \arg\max_{\boldsymbol{a}} Q'(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta}_1)$.
    Perform action $\boldsymbol{a}$, record reward r and next state $\boldsymbol{s}'$;
    Retain sample $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}')$ in the experience replay pool D;
    Update state $\boldsymbol{s} \leftarrow \boldsymbol{s}'$;
    If $|D| > N_r$ ($|D|$ is the number of samples in D), delete the oldest sample;
    If $|D| > N_b$, randomly choose N_b samples $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}')$ from D;
    for $j = 1, \ldots, N_b$
      Select the next action using online network Q_1, $\boldsymbol{a}_j = \arg\max_{\boldsymbol{a}} Q'(\boldsymbol{s}'_j, \boldsymbol{a}_j; \boldsymbol{\theta}_1)$;
      Calculate the direct estimate $Q'(\boldsymbol{s}'_j, \boldsymbol{a}_j; \boldsymbol{\theta}_2)$ of $\boldsymbol{s}'_j$ using the target network Q_2;
      Calculate the indirect estimate $y_j = r_j + \gamma Q'(\boldsymbol{s}'_j, \boldsymbol{a}_j; \boldsymbol{\theta}_2)$;
      Update the parameters of Q_1 $\boldsymbol{\theta}_1^*$ by minimizing $\mathcal{L}(\boldsymbol{\theta}_1) = \frac{\sum_{j=1}^{N_b}\left[\left(y_j - Q'(\boldsymbol{s}_j, \boldsymbol{a}_j; \boldsymbol{\theta}_1)\right)^2\right]}{N_b}$ using stochastic gradient descent;
    end for
  while $\boldsymbol{s} \in \boldsymbol{Z}_w$ or $\tau = N_s$.
  If $\lambda \mod N^- = 0$ ($\mod$ denotes the modulo operation), $\boldsymbol{\theta}_2^* \leftarrow \boldsymbol{\theta}_1^*$.
end for

4. Experiments and Analysis

This section provides various experiments to evaluate the performance of the proposed AMDP-DDQN algorithm in a strategic confrontation game scenario between two different nations (also agents). Specifically, the two agents in the considered scenario are denoted as Country A and Country B, respectively, where each country strategically leverages their strengths to gain advantages and ultimately prevail in the conflict via AMDP strategic confrontation game. For each agent, there are $M = 4$ states consisting of security, economy, technology, and administration.

Furthermore, each agent in the confrontation game can choose one of $K = 3$ types of actions, *i.e.*, attack, defense, and sanction. Each of these actions is subdivided into 10 discrete levels, indexed from 0 to 9. The detailed description of state names, symbols, and their corresponding value ranges are presented in Table 1, where the state values 0, 1, and 2 of the last action type LA_T indicate that the last action type of its opponent is under attack, defense, and sanction, respectively. In addition, we use c_w to denote w-th the game result corresponding to the terminal state $\boldsymbol{Z}_w$, as well as the reward functions for Country A, denoted by $r^A(\boldsymbol{Z}_w)$, and for Country B, denoted by $r^B(\boldsymbol{Z}_w)$. The game result, reward setup, and termination state under different game results are listed in Table 2 where there are a total of 6 game outcomes on the failures of different countries.

The neural network structure used in training and testing is given in Table 3. Specifically, the whole neural network consists of five fully connected layers with ReLU activations after each hidden layer and a linear activation on the output layer. The inputs of the neural network are the state variables listed in Table 1, encompassing the economic, technological, security, and administrative dimensions for both agents at every decision step. The outputs of the neural network are the predicted Q value corresponding to each action, which consists of 30 different action types and action degrees. The hyperparameters of DDQN (including the parameters in epsilon-greedy strategy introduced in Section 4) are given in Table 4.

Table 1. State setting of Agents A and B with confrontation.

State name	State symbol	Value range
Security state of A	M_A	$[0,100]$
Economic state of A	E_A	$[0,100]$
Technological state of A	T_A	$[0,100]$
Administrative state of A	P_A	$[0,100]$
Security state of B	M_B	$[0,100]$
Economic state of B	E_B	$[0,100]$
Technological state of B	T_B	$[0,100]$
Administrative state of B	P_B	$[0,100]$
Last action type	LA_T	{0, 1, 2}
Last action degree	LA_D	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Table 2. Reward setup and termination state under different game results.

Game result c_w	Termination state set Z_w	Reward $r^A(Z_w)$	Reward $r^B(Z_w)$
Security failure of A	$\{s \mid M_A < 0\}$	−20	20
Economic failure of A	$\{s \mid E_A < 0\}$	−30	30
Administration failure of A	$\{s \mid P_A < 0\}$	−50	50
Security failure of B	$\{s \mid M_B < 0\}$	20	−20
Economic failure of B	$\{s \mid E_B < 0\}$	30	−30
Administration failure of B	$\{s \mid P_B < 0\}$	50	−50

Table 3. Structure of the deep neural network in DDQN.

Input: State dimensionality (1 × 10)	
Dense1 + ReLu	10 × 64
Dense2 + ReLu	64 × 256
Dense3 + ReLu	256 × 256
Dense4 + ReLu	256 × 64
Dense5	64 × 30
Output: The predicted Q value for each action (30 × 1)	

Table 4. Training hyperparameter setting of DDQN.

Hyperparameter name	value
Number of episodes	30,000
batch size	64
ϵ_{start}	1
ϵ_{final}	0.001
ϵ_{decay}	2000
N^-	50
N_b	1000

Our experiments, detailed in the subsequent subsections, are conducted on a personal computer equipped with an NVIDIA GeForce RTX 3090 GPU, operating under Ubuntu 22.04. The specific implementation is carried out using Python 3.8, along with the PyTorch 1.12 deep learning framework and the Gym 2.6.0 reinforcement learning library.

4.1. Equal Symmetry Experiment

The equal symmetry experiments are used to simulate a strategic confrontation game between two countries with equal strengths. As a result, the initial state settings of Country A and Country B are kept the same, as shown in **Table 5**.

Table 5. Equal experiment initial state settings.

State symbol	M_A	E_A	T_A	P_A	M_B	E_B	T_B	P_B	LA_T	LA_D
Initial state value	50	50	50	50	50	50	50	50	0	0

The reward evolution curves for Country A and Country B throughout the training process are presented in Figure 5, where "Window size of 10 episodes" denotes the associated curve is obtained by a moving average with a window size of each 10 episodes. As illustrated, both agents (Countries) demonstrate a broadly similar trend: their rewards increase steadily from an initial value close to zero and gradually rise to approximately 25, after which both rewards plateau, indicating a convergence to a stable performance level. This consistent upward trajectory reflects the progressive improvement of agents in policy quality over time, suggesting that both have successfully acquired effective strategies through interaction with the environment. Notably, since the two countries share identical initial state configurations, it is reasonable to expect them to learn comparable optimal policies with a similar game result. In fact, we observe that the convergence of their reward curves further supports the theoretical expectation.

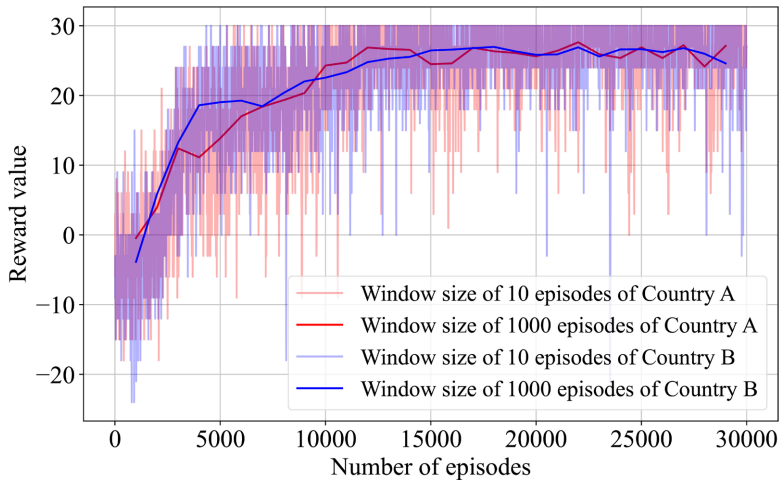


Figure 5. Reward evolution versus the number of episodes during equal symmetry experimental training.

After completing the training phase, we carry out a series of experiments using the trained models of both agents to evaluate the game results of equal symmetry experiment. To evaluate the effectiveness of the learned strategies, four experimental scenarios are designed: 1) both countries adopt random strategies; 2) Country A adopts the model-based strategy while Country B uses a random strategy; 3) Country B adopts the model-based strategy while Country A uses a random strategy; and 4) both countries employ model-based strategies. The associated analysis of the game results is elaborated as follows.

Figure 6 illustrates the state evolution for Country A and Country B over multiple game rounds under different strategy combinations in a symmetric peer-to-peer experimental setting. The dark dashed lines represent the averaged value of

each state across 1000 independent game simulations, while the shaded areas indicate the fluctuation ranges. A state value falling below zero indicates the termination of the game. As shown in Figure 6(a), when both countries adopt random strategies, the evolution of their state values is nearly identical due to the fully symmetric initial conditions, demonstrating the fairness and symmetry of the experimental environment. In contrast, Figure 6(b) shows that when Country A adopts a model-based strategy while Country B keeps using a random strategy, the state values increase significantly, validating the effectiveness of the model-based strategy. A similar outcome is observed in Figure 6(c), where Country B implements the model-based strategy and also experiences a notable improvement in its state values, further highlights the effectiveness and generalizability of the model-based approach. Finally, Figure 6(d) shows that when both countries adopt model-based strategies, a new equilibrium pattern emerges. This indicates that in intelligent strategic games, the mutual adoption of advanced strategies leads to a reestablishment of equilibrium, thereby reaffirming the inherently counterbalancing and adaptive nature of such interactions.

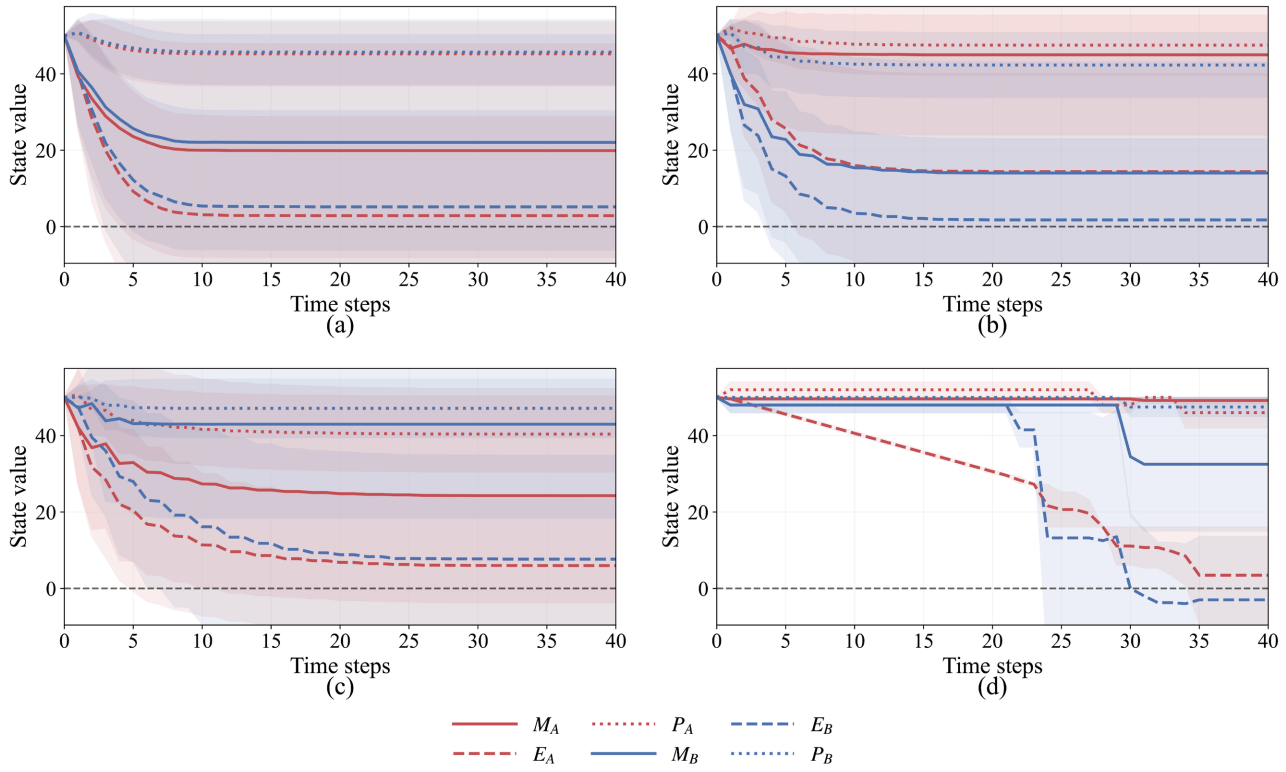


Figure 6. State value evolution over time steps during equal symmetry experiment between Country A and Country B. (a) Both use random strategies; (b) Country A uses a model-based strategy, whereas Country B uses a random strategy; (c) Country B uses a model-based strategy, whereas Country A uses a random strategy; (d) Both use model-based strategies.

The game outcomes of equal symmetry experiment are depicted in Figure 7. The blue bars represent the results when both countries employ random strategies. It is observed that wins and losses are distributed approximately evenly between the two countries, as expected under stochastic behavior. The orange bars

correspond to the scenario where Country A utilizes a model-based strategy while Country B retains a random strategy. Compared to the fully random case, Country A experiences a marked reduction in failure rate, while Country B's failures increase accordingly, indicating the strategic advantage gained from Country A. In contrast, the green bars illustrate the outcomes when Country B adopts a model-based strategy and Country A uses a random one. Here, Country B significantly reduces its failures, while Country A suffers more losses, reflecting the same pattern of strategy advantage. Finally, the red bars depict the case where both countries employ model-based strategies. Given their symmetric initial conditions, the game outcomes converge toward equilibrium, with each country failing approximately half of the time, further validating the competitive balance and mutual adaptation achieved through strategic learning.

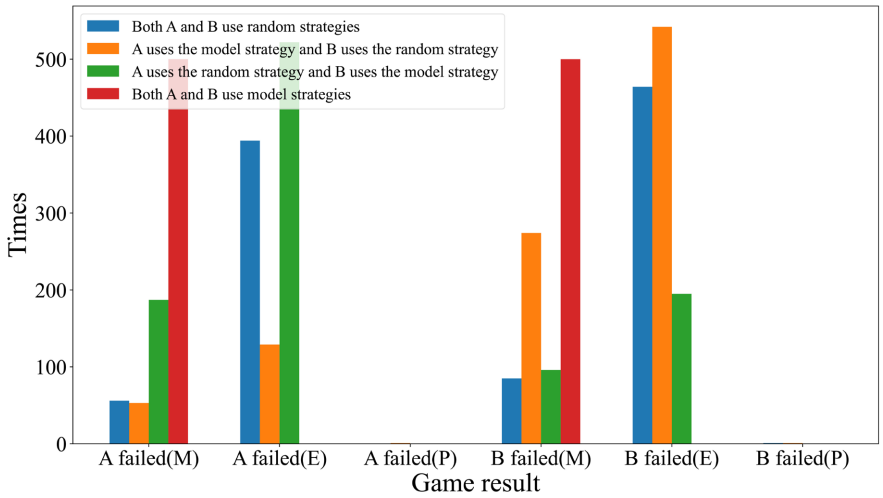


Figure 7. Game results under equal symmetric strategy conditions.

4.2. Non-Equal Experiment

The non-equal experiments are designed to simulate a strategic confrontation between two countries of unequal strengths. In this setting, Country A is assumed to possess a comprehensive advantage against Country B across all aspects in security, economy, technology, and administration. The corresponding initial state configurations for both countries are summarized in Table 6. The reward evolution of each country during training, against the number of episodes, is illustrated in Figure 8. Due to its dominant initial state, the reward curve of Country A quickly rises to approximately 30 and then stabilizes, reflecting rapid convergence to an effective strategy. In contrast, Country B, with disadvantages from its initial conditions, is only able to improve its performance gradually, with the reward curve rising from a negative value to approaching zero over time.

Table 6. Non-equal experiment initial state settings.

State symbol	M_A	E_A	T_A	P_A	M_B	E_B	T_B	P_B	LA_T	LA_D
Initial state value	60	60	60	60	40	40	40	40	0	0

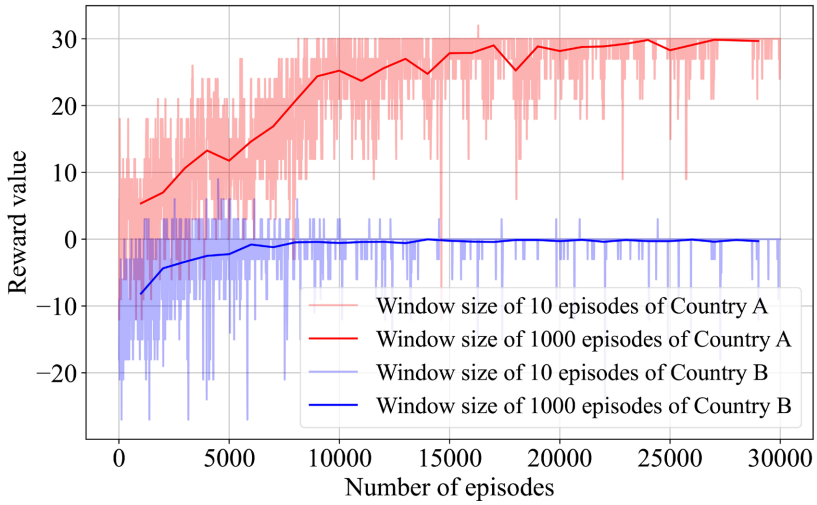


Figure 8. Reward evolution versus the number of episodes during non-equal experimental training.

Figure 9 illustrates the evolution of state values for Country A and Country B under various strategy combinations, beginning from an asymmetric initial condition in which Country B starts with significantly lower state values than Country A. This experimental setting simulates a strategic confrontation scenario where a weaker nation competes against a stronger adversary. In **Figure 9(a)**, both countries adopt random strategies. Due to the considerable initial disadvantage, the state values of Country B fall into the failure region much earlier than that of Country A. This highlights that, under disadvantaged conditions, the weaker side is highly susceptible to rapid suppression if no effective strategy is employed. **Figure 9(b)** presents the case where Country A adopts a model-based strategy while Country B continues using a random strategy. Leveraging its initial advantage, the model of Country A learns to prioritize conservative and risk-averse actions that preserve its dominant state, resulting in consistently high state values and a stable trajectory toward success. This behavior reflects the model's capacity to recognize and maintain strategic superiority through controlled decision-making.

In **Figure 9(c)**, Country B adopts a model-based strategy despite starting from a disadvantaged position. Although its initial state remains inferior, the learned policy enables B to take proactive steps to improve its condition and postpone failure. As a result, the state value of Country B shows significant improvement compared to cases of **Figure 9(a)** and **Figure 9(b)**. This indicates that even from a weak starting point, an effective strategy can prolong engagement and create potential opportunities. **Figure 9(d)** examines the scenario where both countries deploy model-based strategies. Despite the presence of strategic reasoning on both sides, the initial advantages of Country A enable its model to quickly identify and exploit this asymmetry by adopting an aggressive optimal policy. Consequently, Country B, although supported by a model, fails to attain an effective counter-strategy and suffers a swift decline in state values. This result illustrates a dominant amplification effect, where model-based strategies not only reinforce but in-

tensify the impact of favorable initial conditions, allowing the dominant player to dictate the pace of the confrontation game and suppress any potential actions of the weaker side.

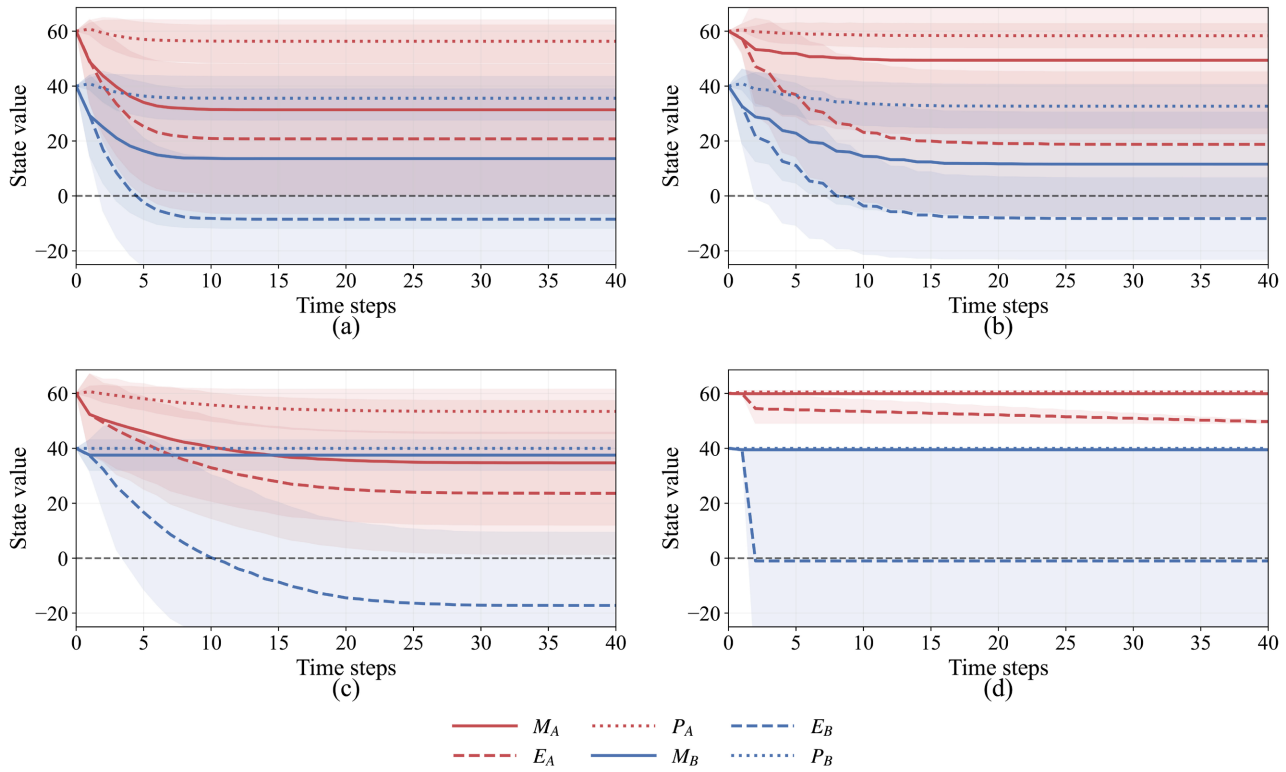


Figure 9. State value evolution over time steps during non-equal experiment between Country A and Country B. (a) Both use random strategies; (b) Country A uses a model-based strategy, whereas Country B uses a random strategy; (c) Country B uses a model-based strategy, whereas Country A uses a random strategy; (d) Both use model-based strategies.

Overall, the results of **Figure 9** demonstrate how initial conditions, when coupled with strategic learning, can significantly influence the dynamics of adversarial interactions. This may lead to irreversible trajectories shaped by early asymmetries.

The game results shown in **Figure 10** reveal clear outcome disparities under different strategy combinations. When both countries adopt random strategies, Country A wins the majority of the games, owing to its initial advantage. This disparity becomes even more pronounced when Country A employs the model-based strategy, leading to an overwhelming dominance in which it secures nearly all victories. In contrast, when Country B adopts the model-based strategy while Country A uses a random strategy, the improvement in Country B's performance is marginal. Despite the strategic upgrade, the significant disadvantage in its initial state limits the effectiveness of the model, resulting in only slightly better outcomes compared to using a random strategy. Finally, when both countries utilize model-based strategies, Country A still wins all the games, solely due to its strong initial advantage. This outcome underscores that, in highly asymmetric settings,

strategic sophistication alone may be insufficient to overcome substantial disparities in starting conditions.

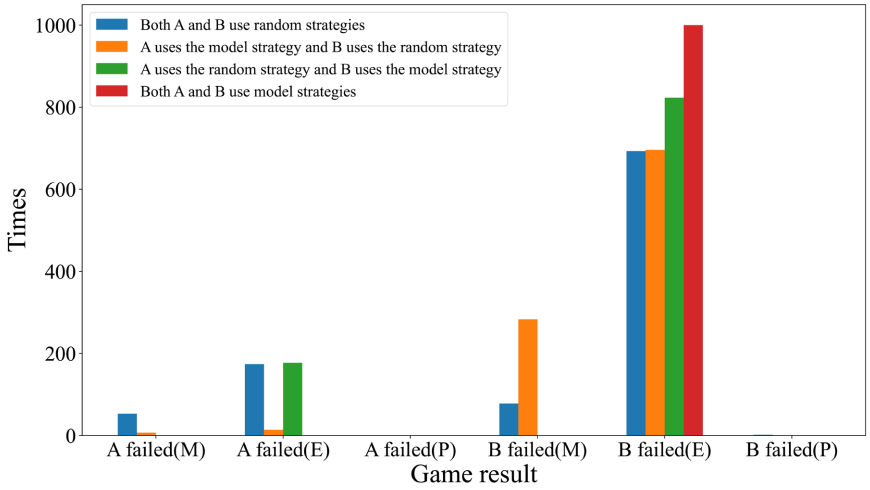


Figure 10. Game results under non-equal symmetric strategy conditions.

4.3. Equal Asymmetry Experiment

Finally, we validate the performance of the proposed AMDP-DDQN under the equal asymmetry condition, *i.e.*, the two countries with roughly the same total strength but with different state values. Specifically, Country A is assigned stronger security power but a weaker economy, while Country B has the inverse situation: a stronger economy and weaker security power. The corresponding initial state configurations are detailed in Table 7. The reward evolution versus the number of episodes during equal asymmetry experiment is depicted in Figure 11, where the average reward in training of both countries is gradually rising and stable, in which country A is stable around 25 while country B is stable around 20. It is seen that under equal asymmetry conditions, both countries gradually learn effective strategies, with rewards stabilizing after initial fluctuations. Country A consistently achieves higher rewards than those of Country B, suggesting that the security advantage has a stronger impact on the outcomes. This indicates that even with equal total strength, the equal asymmetry condition significantly influences the final decision-making performance.

Figure 12 illustrates the state evolution of Country A and Country B under four strategy combinations in an equal-but-asymmetric setting, where we keep the same strategy combinations as in the previous experiments. In this experiment, both countries start with the same initial state value, but with different internal distributions of respective states: Country A begins with a relatively weaker economic state, while Country B is more vulnerable in its security domain.

Table 7. Equal asymmetry experiment initial state settings.

State symbol	M_A	E_A	T_A	P_A	M_B	E_B	T_B	P_B	LA_T	LA_D
Initial state value	80	40	50	50	40	80	50	50	0	0

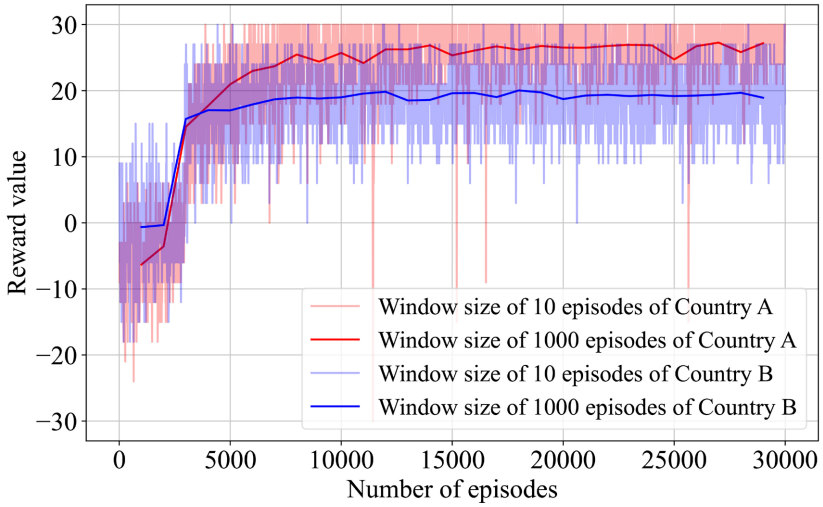


Figure 11. Reward evolution versus the number of episodes during equal asymmetry experimental training.

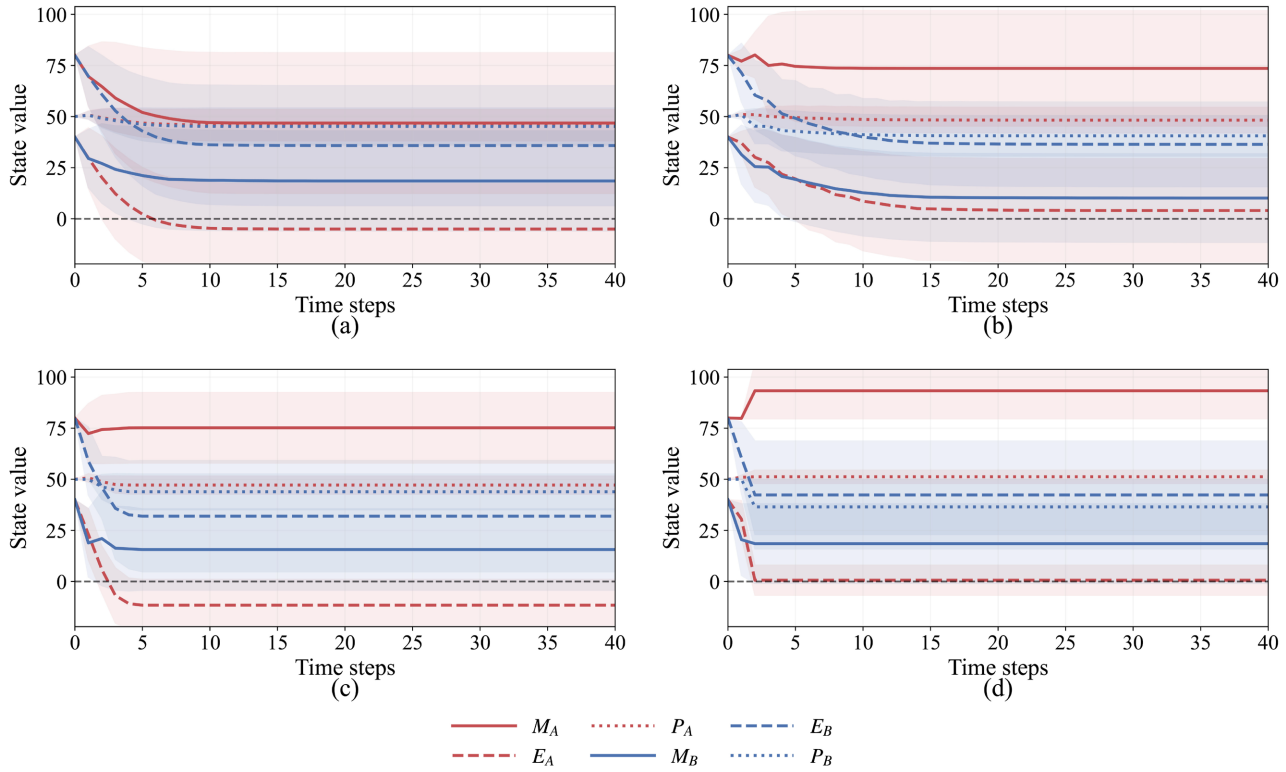


Figure 12. State value evolution over time steps during equal asymmetry experiment between Country A and Country B. (a) Both use random strategies; (b) Country A uses a model-based strategy, whereas Country B uses a random strategy; (c) Country B uses a model-based strategy, whereas Country A uses a random strategy; (d) Both use model-based strategies.

In the scenario of Figure 12(a), both countries adopt random strategies. Due to Country A's initial economic weakness, its overall development capability remains limited, leading to a rapid decline in state value and a higher risk of failure. This indicates that even with equal total state values, structural weaknesses in some crucial initial state can enlarge the risk of defeat. In Figure 12(b), Country

A instead adopts a model-based strategy. The well-trained model of Country A first focuses on improving its economic state to ensure long-term sustainability, while also identifying and exploiting Country B's security weakness. This dual strategy helps Country A delay its decline and obtain a temporary advantage, thereby demonstrating the adaptability and strategic precision of the training model in scenarios with asymmetric resource distribution.

In Figure 12(c), Country B employs a model-based strategy. The training model of Country B accurately identifies the economic vulnerability of Country A, resulting in rapid defeat of Country A. This result shows that the training model can clearly recognize the main weakness of its opponent and respond with effective and targeted strategies. In Figure 12(d), both countries adopt model-based strategies. In this experiment, both countries have their own weaknesses, but the economic weakness of Country A is more serious than the security weakness of Country B. As a result, when both countries use their trained model strategies to attack the weaknesses of the other one, Country A is more likely to be defeated first because its weakness is more critical.

In equal asymmetry experiment, both countries start with the same initial state value, but with different internal distributions of respective states. That is to say, Country A begins with a relatively weaker economic state, while Country B is more vulnerable in its security domain. We conclude that the equal-but-asymmetric setup reflects real-world situations where two agents with confrontation game have similar overall strength but differ in their resource distribution, which also leads to different strategic outcomes.

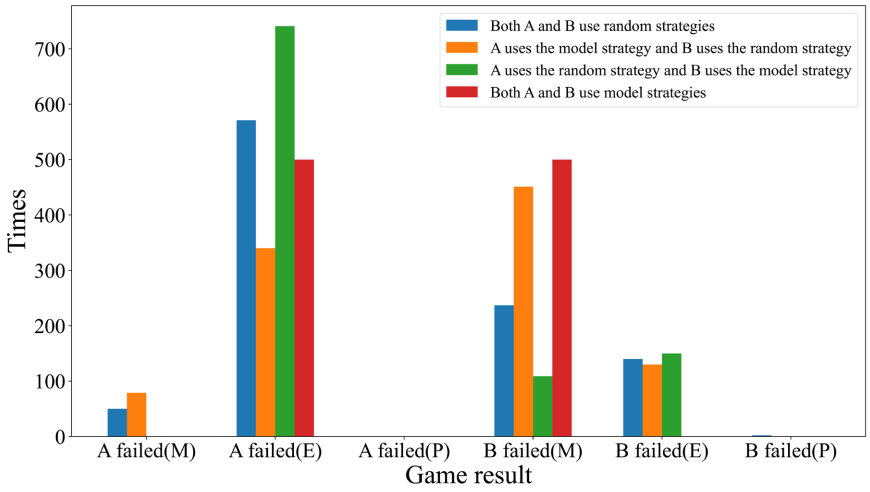


Figure 13. Game results under equal asymmetry conditions.

We now illustrate the game results of the equal asymmetry experimental games in Figure 13. Note that when both agents (countries) adopt random strategies, while Country A has more economic failure than that of Country B and Country B has more security failure than that of Country A. This aligns with our initial state hypothesis that Country A has a weaker economic state and Country B has a

weaker security state. In addition, when Country A uses the model strategy and Country B uses the random strategy, the game results show that the number of economic failure of Country A is greatly reduced while the number of security failure of Country B is greatly increased. This implies that Country A makes up for its economic weakness through actions while attacking the security weakness of Country B. Similarly, when Country B uses model strategy and Country A uses random strategy, the number of security failure of Country B decreases while the number of economic failures of Country A increases. This indicates that Country B also mitigates its own security weakness through effective actions of attacking the economic weakness of the other one. Finally, when both agents use the model strategy, we see that all of game outcomes exhibit failures due to their own weaknesses, and the number of failures of both countries are roughly equivalent.

To further investigate the model-based strategies which are learned from equal asymmetry training, the actions with the highest Q-values under different states are visualized using three-dimensional plots, as shown in Figure 14 and Figure 15, where different colors represent different types of actions, and the color intensity indicates the action degree. In these figures, the x- and y-axes represent the current state levels of Country A and Country B, respectively, where each axis value denotes a uniform state setting across all four dimensions of the corresponding country. The z-axis indicates the opponent's action in the previous round. It is observed from the two figures that Country A tends to choose defensive (green) actions most of the time, resorting to offensive or sanction actions only when the opponent is in a weakened state. In contrast, Country B often prefers sanction actions, leveraging its stronger economic position to gain an advantage.

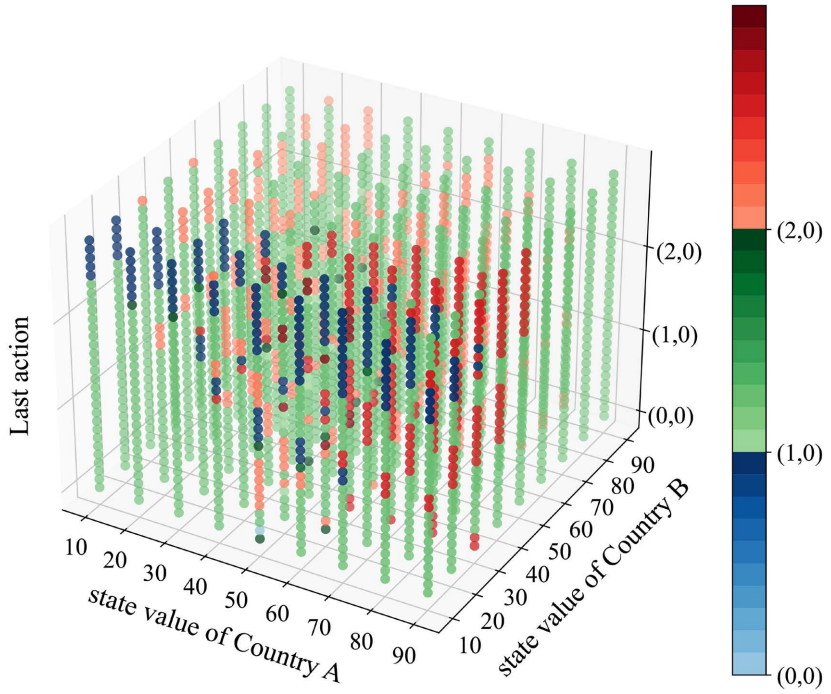


Figure 14. The maximum Q-value action distribution of Country A with different states.

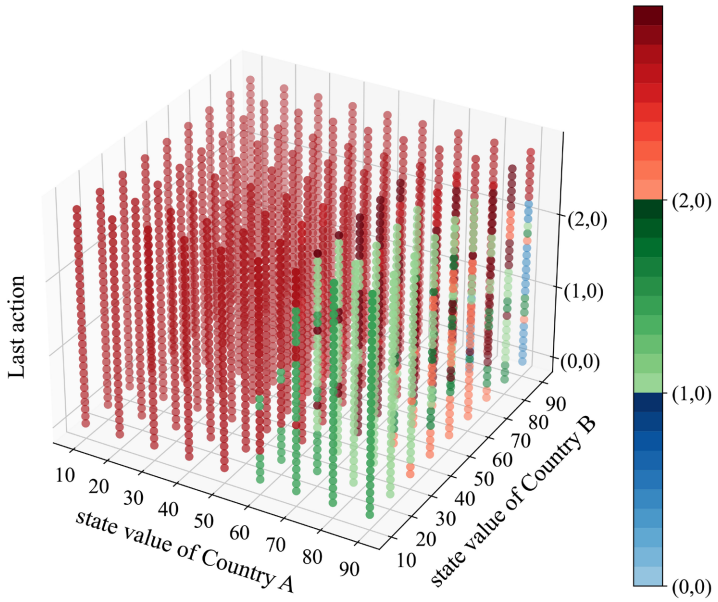


Figure 15. The maximum Q-value action distribution of Country B with different states.

Overall, all above experimental results clearly show that the AMDP-DDQN-trained models are able to make smart decisions and choose proper actions by considering both the current state and the previous action of the corresponding opponent. This allows each agent to adjust its strategy and respond effectively in different situations. Whatever facing symmetric or asymmetric scenarios, the associated models consistently choose actions that improve the chances of winning. Compared with random strategies, the model-based strategies usually lead to significantly superior outcomes, including higher rewards and longer survival times. This validates that the proposed AMDP-DDQN approach can successfully learn useful and adaptive policies, and that these well-learned strategies are effective across a variety of game settings.

5. Conclusion

In this paper, we have introduced an alternating Markov decision process (AMDP) to model the sequential and strategic interactions between multi-agents with confrontation game. The proposed AMDP approach captures the sequential and interdependent decision-making dynamic characteristic of complex strategic environments. Meanwhile, we have proposed the AMDP-DDQN training algorithm for the multi-agent strategic confrontation game based on the double deep Q-learning. Meanwhile, we also exhibited extensive experiment results in a strategic confrontation game scenario between two countries with strategic confrontation to demonstrate the effectiveness and generalizability of the proposed AMDP-DDQN approach.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Han, S., Ke, L. and Wang, Z. (2021) Multi-Agent Confrontation Game Based on Multi-Agent Reinforcement Learning. 2021 *IEEE International Conference on Unmanned Systems* (*ICUS*), Beijing, 15-17 October 2021, 157-162. https://doi.org/10.1109/icus52573.2021.9641171

[2] Marwala, T. (2023) Artificial Intelligence, Game Theory and Mechanism Design in Politics. Springer Nature.

[3] Fudenberg, D. and Tirole, J. (1991) Game Theory. MIT Press.

[4] Wellman, M.P., Tuyls, K. and Greenwald, A. (2025) Empirical Game Theoretic Analysis: A Survey. *Journal of Artificial Intelligence Research*, **82**, 1017-1076. https://doi.org/10.1613/jair.1.16146

[5] Von Neumann, J. and Morgenstern, O. (2007) Theory of Games and Economic Behavior: 60th Anniversary Commemorative Edition. Princeton University Press.

[6] Roth, A.E. and Erev, I. (1995) Learning in Extensive-Form Games: Experimental Data and Simple Dynamic Models in the Intermediate Term. *Games and Economic Behavior*, **8**, 164-212. https://doi.org/10.1016/s0899-8256(05)80020-x

[7] Shi, Q. (2024) Responsibility in Extensive Form Games. *Proceedings of the AAAI Conference on Artificial Intelligence*, **38**, 19920-19928. https://doi.org/10.1609/aaai.v38i18.29968

[8] Harsanyi, J.C. (1973) Games with Incomplete Information Played by "Bayesian" Players, I-III Part I. The Basic Model. *Management Science*, **14**, 159-182. https://doi.org/10.1038/246015a0

[9] Smith, J.M. and Price, G.R. (1973) The Logic of Animal Conflict. *Nature*, **246**, 15-18. https://doi.org/10.1038/246015a0

[10] Davis, P.K. and Winnefeld, J.A. (1983) The RAND Strategy Assessment Center: An Overview and Interim Conclusions about Utility and Development Options. Rand Corporation. https://www.rand.org/pubs/reports/R2945.html

[11] Bartels, E.M. (2020) Building Better Games for National Security Policy Analysis. PhD Dissertation, Pardee RAND Graduate School. https://www.rand.org/content/dam/rand/pubs/rgs_dissertations/RGSD400/RGSD437/RAND_RGSD437.pdf

[12] Davis, P.K. and Bracken, P. (2022) Artificial Intelligence for Wargaming and Modeling. *The Journal of Defense Modeling and Simulation*: *Applications*, *Methodology*, *Technology*, **22**, 25-40. https://doi.org/10.1177/15485129211073126

[13] Bellman, R. (1966) Dynamic Programming. *Science*, **153**, 34-37. https://doi.org/10.1126/science.153.3731.34

[14] Kaelbling, L.P., Littman, M.L. and Cassandra, A.R. (1998) Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*, **101**, 99-134. https://doi.org/10.1016/s0004-3702(98)00023-x

[15] Nowé, A., Vrancx, P. and De Hauwere, Y. (2012) Game Theory and Multi-Agent Reinforcement Learning. In: Wiering, M. and Otterlo, M., Eds., *Adaptation*, *Learning*, *and Optimization*, Springer, 441-470. https://doi.org/10.1007/978-3-642-27645-3_14

[16] Yao, Q., Wang, Y., Xiong, X., Wang, P. and Li, Y. (2023) Adversarial Decision-Making for Moving Target Defense: A Multi-Agent Markov Game and Reinforcement Learning Approach. *Entropy*, **25**, Article No. 605. https://doi.org/10.3390/e25040605

[17] Ning, Z. and Xie, L. (2024) A Survey on Multi-Agent Reinforcement Learning and Its Application. *Journal of Automation and Intelligence*, **3**, 73-91.

https://doi.org/10.1016/j.jai.2024.02.003

[18] Chi, P., Wei, J., Wu, K., Di, B. and Wang, Y. (2023) A Bio-Inspired Decision-Making Method of UAV Swarm for Attack-Defense Confrontation via Multi-Agent Reinforcement Learning. *Biomimetics*, **8**, Article No. 222. https://doi.org/10.3390/biomimetics8020222

[19] Zeng, Q. and Nait-Abdesselam, F. (2024) Multi-Agent Reinforcement Learning-Based Extended Boid Modeling for Drone Swarms. *ICC* 2024—*IEEE International Conference on Communications*, Denver, 9-13 June 2024, 1551-1556. https://doi.org/10.1109/icc51166.2024.10622479

[20] He, R., Wu, D., Hu, T., Tian, Z., Yang, S. and Xu, Z. (2024) Intelligent Decision-Making Algorithm for UAV Swarm Confrontation Jamming: An M2AC-Based Approach. *Drones*, **8**, Article No. 338. https://doi.org/10.3390/drones8070338

[21] Foerster, J., Assael, I.A., De Freitas, N. and Whiteson, S. (2016) Learning to Communicate with Deep Multi-Agent Reinforcement Learning. *International Conference on Neural Information Processing Systems* (*NeurIPS*), Barcelona, 5-10 December 2016, 29. https://doi.org/10.5555/3157096.3157336

[22] Arulkumaran, K., Deisenroth, M.P., Brundage, M. and Bharath, A.A. (2017) Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine*, **34**, 26-38. https://doi.org/10.1109/msp.2017.2743240

[23] François-Lavet, V., Henderson, P., Islam, R., Bellemare, M.G. and Pineau, J. (2018) An Introduction to Deep Reinforcement Learning. *Foundations and Trends® in Machine Learning*, **11**, 219-354. https://doi.org/10.1561/2200000071

[24] Standen, M., Kim, J. and Szabo, C. (2025) Adversarial Machine Learning Attacks and Defences in Multi-Agent Reinforcement Learning. *ACM Computing Surveys*, **57**, 1-35. https://doi.org/10.1145/3708320

[25] Demura, Y. and Kaneko, T. (2024) Initial State Diversification for Efficient Alphazero-Style Training. *ICGA Journal*, **46**, 40-66. https://doi.org/10.3233/icg-240255

[26] Mnih, V., Kavukcuoglu, K., Silver, D., *et al*. (2013) Playing Atari with Deep Reinforcement Learning. https://arxiv.org/abs/1312.5602.

[27] Van Hasselt, H., Guez, A. and Silver, D. (2016) Deep Reinforcement Learning with Double Q-Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, **30**, 2094-2100. https://doi.org/10.1609/aaai.v30i1.10295

[28] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D. and Riedmiller, M. (2014) Deterministic Policy Gradient Algorithms. *International Conference on Machine Learning* (*ICML*), Beijing, 22-24 June 2014, 387-395. http://proceedings.mlr.press/v32/silver14.pdf

[29] Liu, P., Cui, H. and Zhang, N. (2025) DDQN-Based Centralized Spectrum Allocation and Distributed Power Control for V2X Communications. *IEEE Transactions on Vehicular Technology*, **74**, 4408-4418. https://doi.org/10.1109/tvt.2024.3493137

[30] Zhang, K., Li, Y., Zhang, Z. and Ye, F. (2025) DDQN-Based Hybrid Routing Protocol for UWSNs with Void Repair Mechanism. *IEEE Sensors Journal*, **25**, 20718-20731. https://doi.org/10.1109/jsen.2025.3557067

[31] Lu, S., Tao, Y., Zeng, J. and Zuo, Q. (2025) A Study on the Impact of Obstacle Size on Training Models Based on DQN and DDQN. *ITM Web of Conferences*, **73**, Article No. 01004. https://doi.org/10.1051/itmconf/20257301004

[32] Yan, Z., Zhou, H., Tabassum, H. and Liu, X. (2025) Hybrid LLM-DDQN-Based Joint Optimization of V2I Communication and Autonomous Driving. *IEEE Wireless Communications Letters*, **14**, 1214-1218. https://doi.org/10.1109/lwc.2025.3539638

[33] Yu, X., Jiang, J. and Lu, Z. (2024) Opponent Modeling Based on Subgoal Inference. 38*th Conference on Neural Information Processing Systems* (*NeurIPS* 2024), Vancouver, 9-15 December 2024, 60531-60555. https://proceedings.neurips.cc/paper_files/paper/2024/file/6fb9ea5197c0b8ece8a64220fb82cdfe-Paper-Conference.pdf

[34] Kingma, D.P. and Ba, J. (2014) Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (*ICLR*), San Diego, 7-9 May 2015. https://arxiv.org/abs/1412.6980