Journal of *Marine Science and Engineering*

MDPI

Article

Intelligent Ship Collision Avoidance Algorithm Based on DDQN with Prioritized Experience Replay under COLREGs

Pengyu Zhai, Yingjun Zhang * and Wang Shaobo

Navigation College, Dalian Maritime University, Dalian 116026, China; zhai_pengyu@dlmu.edu.cn (P.Z.); wangshaobo1@163.com (W.S.)
* Correspondence: zhangyj@dlmu.edu.cn

Abstract: Ship collisions often result in huge losses of life, cargo and ships, as well as serious pollution of the water environment. Meanwhile, it is estimated that between 75% and 86% of maritime accidents are related to human factors. Thus, it is necessary to enhance the intelligence of ships to partially or fully replace the traditional piloting mode and eventually achieve autonomous collision avoidance to reduce the influence of human factors. In this paper, we propose a multi-ship automatic collision avoidance method based on a double deep Q network (DDQN) with prioritized experience replay. Firstly, we vectorize the predicted hazardous areas as the observation states of the agent so that similar ship encounter scenarios can be clustered and the input dimension of the neural network can be fixed. The reward function is designed based on the International Regulations for Preventing Collision at Sea (COLREGs) and human experience. Different from the architecture of previous collision avoidance methods based on deep reinforcement learning (DRL), in this paper, the interaction between the agent and the environment occurs only in the collision avoidance decision-making phase, which greatly reduces the number of state transitions in the Markov decision process (MDP). The prioritized experience replay method is also used to make the model converge more quickly. Finally, 19 single-vessel collision avoidance scenarios were constructed based on the encounter situations classified by the COLREGs, which were arranged and combined as the training set for the agent. The effectiveness of the proposed method in close-quarters situation was verified using the Imazu problem. The simulation results show that the method can achieve multi-ship collision avoidance in crowded waters, and the decisions generated by this method conform to the COLREGs and are close to the level of human ship handling.

Keywords: collision avoidance; reinforcement learning; DDQN with prioritized experience replay; COLREGs; intelligent ship

Citation: Zhai, P.; Zhang, Y.; Shaobo, W. Intelligent Ship Collision Avoidance Algorithm Based on DDQN with Prioritized Experience Replay under COLREGs. *J. Mar. Sci. Eng.* **2022**, *10*, 585. https://doi.org/10.3390/jmse10050585

Academic Editors: Jacopo Aguzzi and Daniel Mihai Toma

Received: 4 April 2022
Accepted: 22 April 2022
Published: 26 April 2022

1. Introduction

As global economic activities become more interconnected, the density of maritime traffic flows is increasing, especially in inshore navigation and in fishing areas. Clearly, this situation increases the risk of collisions between vessels. Ship collisions often result in significant casualties and economic damage. At present, the officer on watch (OOW) can obtain navigational data on surrounding vessels through navigation aids, such as radar and automatic identification systems (AIS). However, contrary to expectations, misinterpretation or omission of information by the pilot can sometimes lead to incorrect decisions or untimely action. Surveys show that approximately 94.7% of ship-to-ship collisions in the last 43 years have been caused by human error on the part of the crew, and at least 56% of collisions are caused by violations of the International Regulations for Collision Avoidance at Sea (COLREGs) established by the International Maritime Organization (IMO) [1,2]. Therefore, the development of autonomous collision avoidance systems that comply with navigation rules is one of the most effective ways to reduce the human factor and the incidence of collisions by improving the intelligence of ships.

In recent years, with the continuous improvement of the theory of unmanned control technology and the gradual development of perception systems, unmanned vessels have started to gradually replace human operators in performing tasks. Unmanned surface vehicles (USVs), as small unmanned offshore platforms, have a wide range of applications in both military and civilian fields, such as port patrol, coast guard, environmental monitoring, seaway mapping, etc. [3]. Maritime autonomous surface ship (MASS) is considered to be an important development field in maritime transportation systems. Both USV and MASS should develop autonomous navigation systems. Moreover, the key technology of autonomous navigation systems is the autonomous collision avoidance technology to ensure safety of navigation. The development of unmanned ships is in its infancy, which means that in the long term future, there will be a coexistence of manned and unmanned ships. In order to adapt to this situation, the developed automatic collision avoidance system should be in line with the current collision avoidance rules and human habits, although the COLREGs may be modified because of MASS operation [4].

The problem with autonomous collision avoidance on ships is one of decision optimization that has long been of interest to scholars. Before the rapid development of machine learning, scholars proposed many deterministic and heuristic algorithms, such as fuzzy control, the artificial potential field method (APF), the velocity obstacle method (VO), the A*-based global path planning collision avoidance algorithm, Dijkstra path planning, GA (genetic algorithm)-based collision avoidance algorithm, etc. [5–12]. However, these algorithms have their limitation; for example, the APF algorithm cannot be applied in complex and uncertain navigation environment, and the algorithm itself has the defect of easily falling into local minima. The A* algorithm relies on a grid map and is very effective for avoiding static obstacles but is not suitable for dynamic environments, with the generated paths needing to be smoothed. Due to the problems of model-dependent accuracy, grid maps and computational complexity of traditional algorithms, artificial intelligence techniques have been applied to ship collision avoidance decision making in recent years with the rapid development of machine learning, especially deep reinforcement learning tools. Reinforcement learning (RL) is an important branch of artificial intelligence in which learning takes place through interaction with the environment. It is a goal-oriented form of learning where the learner is not told what behavior to execute but instead learns from the consequences of their actions. The ship collision avoidance decision problem is a typical Markov decision process (MDP), so reinforcement learning can be used to solve this problem. Deep reinforcement learning (DRL) has the unique advantage that it does not rely on model building but on state transition information gathered through interaction with the environment, bypassing complex issues, such as system modelling, and using the ability of deep neural networks (DNNs) to fit non-linear systems to achieve sequential decisions. The agent is allowed to learn offline, which means that real human collision avoidance experience can be learned so that agents make decisions that are more like human behavior. The agent makes collision avoidance decisions like human behavior by learning real human collision avoidance experience by offline learning. Therefore, a collision avoidance algorithm based on an improved deep Q network (DQN) is proposed in this paper. This method can produce collision avoidance actions that comply with COLREGs and are similar to human maneuvering.

2. Literature Review

The issue of ship automatic collision avoidance has, so far, attracted the attention of many researchers, and relevant theories and technologies are constantly being updated and developed. Numerous collision avoidance algorithms can be broadly classified into two categories: traditional methods based on modelling the environment (model-based) and deep reinforcement learning algorithms that are model-free.

In the field of traditional model-based algorithms, the A* algorithm generates the optimal path by minimizing the path cost. Liu et al. [8] proposed an improved A* algorithm that takes into account collision avoidance rules and ship maneuverability to automatically gen-

erate optimal paths that combine economy and safety. Yogang et al. [9] used a constrained A* method for global path planning with the simultaneous inclusion of static obstacles, dynamic obstacles and currents of different intensities and also investigated the effect of wind currents on optimal waypoints in selected environments. Ship collision avoidance is an uncertain system of interaction between humans, the ship and the marine environment, with time-varying, non-linear characteristics, so some scholars apply fuzzy set to the field of ship collision avoidance. Namgung et al. [13] developed a collision risk assessment system based on fuzzy logic, which calculates the CRI (collision risk index) to determine the optimal time and distance to take collision avoidance action. Some scholars also use model predictive control (MPC) to avoid collisions. Xie et al. [14] proposed an MPC method combined with improved Q-learning beetle swarm search (IQ-BSAS) and neural networks. This algorithm achieves collision avoidance by using neural networks to approximate the inverse model of MPC decisions. Multi-objective evolutionary algorithms (MOEAs) are a class of global probabilistic optimization search methods that mimic biological evolutionary mechanisms to make multiple objectives as optimal as possible for a given region at the same time. Li et al. [15] used the improved non-dominated sorting genetic algorithm II (NSGA-II) and ship domain model to optimize the ship collision avoidance strategy by balancing the safety and economy of ship collision avoidance actions. Zhou Kang et al. [16] proposed a method to reduce fuel consumption during collision avoidance, but their simulation results show that the model outputs a large rudder angle for small changes of course, which is a clear departure from sailing practice. Ni et al. [10] used multiple genetic algorithms and linear expansion algorithms for planning collision avoidance trajectories. Liu et al. [17] adopted a hybrid optimization algorithm combining an improved bacterial foraging algorithm and a particle swarm optimization algorithm to optimize the ship's collision avoidance path and generated the optimal collision avoidance steering angle. The path planning of the artificial potential field (APF) method is a virtual force method proposed by Khatib. Its basic idea is to design an abstract gravitational field. The target point produces "gravity" on the ship, the obstacle ship produces "repulsion" and, finally, by seeking the resultant force, to control the trajectory of the ship's movement. The paths planned by the potential field method are generally smooth and safe, but this method has the problem of local optimum. Li et al. [18] used the APF to improve the action space and reward function of the DQN algorithm, solved the problem of sparse reward function in the reinforcement learning algorithm and realized the path planning of autonomous collision avoidance. Hongguang Lyu and Yong Yin [5] proposed a path-oriented hybrid artificial potential field method (PGHAPF) that has the potential to integrate potential field models into ECDIS for path planning and is extremely feasible for practical engineering applications. The VO algorithm proposed by Fiorini and Shiller [19] could resolve conflicts with multiple moving obstacles, and this algorithm collects all the velocities that result in collisions and presents a set of collision-free velocities, which facilitates a machine search for the best option. Huang et al. [7] applied the generalized velocity obstacle (GVO) method to the field of multi-ship collision avoidance, and the simulation results show that the algorithm can take fewer necessary actions. Shaobo et al. [6] developed an autonomous navigation system based on an improved VO algorithm combined with a finite state machine (FSM), introducing a multi-level optimization decision model considering ship maneuverability, COLREGs and other constraints. A case study was carried out on ECDIS, the results of which show the robustness of this system under various sea conditions.

Model-free RL algorithms are well adapted to complex systems and have self-learning capabilities that discover optimal strategies from unknown environments through trial-and-error interactions, thus providing an effective way to deal with extremely complex systems. There are two main branches of DRL. One is the method of learning the optimal action value function, such as deep Q networks (DQNs), which combine value-based Q-learning with neural networks, using neural networks to approximate the action value of all possible behavior in each state, effectively solving the problem that Q-Learning can only make decisions in an environment with a limited number of states. The other is the DRL

algorithm, which is based on policy gradients, such as DDPG, which can handle continuous action spaces. These methods are often used in the field of ship collision avoidance to control the rudder angle of a ship. Both types of algorithms have been widely used in the field of ship collision avoidance. Shen et al. [20] proposed an intelligent collision avoidance algorithm based on DQN, which applies the ship domain and predicts area of danger to describe the area of obstruction, uses 12 distance detection lines to detect the distance to the area of obstruction and uses the distance at five consecutive moments as the input to the DQN. Numerical simulations were carried out in conjunction with a ship model, and, finally, pool experiments are conducted using three scaled models of ships with different maneuverability to verify the possibility of intelligent collision avoidance in more complex waters. Zhao and Roh [21] proposed an actor–critic (AC) algorithm for ship collision avoidance. This algorithm divides the target ship area into four regions based on COLREGs, solving the problem of fixing the input dimension of the neural network when the number of target ships varies. This method considers only the nearest ship in the divided region and does not consider all ships and static obstacles. Sawada et al. [22] proposed a collision avoidance method based on proximal policy optimization (PPO). This method improves obstacle zone by target (OZT), uses grid sensor to quantize OZTs and uses A convolutional neural network (CNN) and long-short-term memory network (LSTM) to control the rudder angle in the continuous action space. This method is only suitable for open waters. When the number of obstructions and target ships is increased, the number and area of OZTs increase, which may cause the action taken to fail to change the observed state of the agent. Jiahui Shi [23] first mined successful collision avoidance cases in Tianjin port AIS big data and used a double GRU-RNN network for learning, which provides a new solution for massive AIS data in a practical application. Chen et al. [24] proposed a collision avoidance method based on Q-learning for path planning, which allowed the ship to sail independently along the appropriate path or navigation strategy without human intervention. Compared with the traditional path planning method, the effect of this algorithm is closer to that of human operation. Xu et al. [25] proposed a COLREGs intelligent collision avoidance (CICA) algorithm, employing DNN to automatically extract the observation state characteristics and proposing a new method to track the current network weight to update the target network weight, which improves the stability of learning the optimal strategy. Compared with other updating strategies, their training convergence time was significantly reduced. Finally, the simulation results were compared with VO and APF algorithms, and the results were obviously better than those of the other two algorithms. Woo and Kim [26] proposed a USV collision avoidance algorithm based on improved DQN, which represented the encounter situation of ships with a grid map and used the visual recognition capabilities of DNN to realize the perception of the encounter situation of ships and collision avoidance decision making. A VO algorithm was used to plan the collision avoidance path. In contrast to previous decisions, this network only decides whether to take collision avoidance action or continue to follow the route. If an action should be taken, steering to port or starboard is decided by the DQN, and the steering angle is generated by the VO algorithm. However, this approach is a departure from the basic idea of reinforcement learning.

The above studies were conducted from the perspective of safety and compliance with collision avoidance rules and did not considered whether collision avoidance decisions are consistent with human maneuvering habits, with trajectories that are difficult to track, frequent steering and the use of large rudder angles (close to full rudder) to shift very small courses. Moreover, a ship's collision avoidance usually consists of four parts: environment perception, taking collision avoidance action, keeping course and speed, and returning to the planned route. Usually, a complete collision avoidance process will take ten minutes or more, and the stage of keeping course and speed, as well as returning to the planned route, takes up most of the time, whereas taking collision avoidance action is a relatively short step. The decision to take action is the core part of collision avoidance. In most cases, it takes a few actions to make the collision risk disappear. Therefore, using a reinforcement

learning algorithm for the entire collision avoidance process increases the length of the state transition chain, which leads to an increase in the complexity of the algorithm and, consequently, to learning difficulties. In addition, the RL algorithm relies heavily on the input of the observation state, which directly affects the learning ability of the agent. In previous studies, ship kinematic parameters, distance and bearing have usually been used as inputs, but raw, unprocessed data are difficult to learn and often require a deeper and wider network structure to process. At the same time, the number of target ships and the number of static obstacles may result in different dimensions of the observed states.

Motivated by all of the above, we designed a novel method for collision avoidance based on DRL. In this paper, we proposed a new strategy that uses dynamic information about obstructions and target vessels to predict hazardous areas for collisions and quantified the areas as observation states of the agent to fix the input dimension and cluster similar collision avoidance scenarios. The DRL algorithm is applied only to the collision avoidance decision-making phase so that the number of state transitions in the collision avoidance process can be reduced to optimize the learning process while using prioritized experience replay to accelerate the learning process. In addition, the reward function is designed to take into account the COLREGs and human maneuvering habits so that the agent can take collision avoidance actions, effectively comply with the collision avoidance rules and more closely mimic human behavior. Finally, a simulation of collision avoidance in a crowded water environment with multiple vessel encounters was carried out.

To summarize the above, the contributions of this paper are concluded as follows:

1. A novel automatic collision avoidance strategy is proposed. The strategy applies the DRL navigation algorithm only in the collision avoidance decision-making phase, which reduces the number of state transitions in the Markov decision process and thus optimizes the learning process. The convergence rate is also optimized by using an improved DQN.
2. Based on the perception information, hazardous areas are predicted and quantified as input to the DRL algorithm, which serves to fix the dimensionality of the observation state of the agent and to cluster similar ship collision avoidance scenarios.
3. When designing the reward function for DRL, COLREGs and human experience are taken into account so that the decisions made by the agent more closely resemble human operation.
4. Nineteen single-ship encounter scenarios (non-close-quarters situations) were created based on the ship encounter situations classified by the COLREGs. These scenarios were arranged and combined to serve as the training set for the agent. The Imazu problem, a set of more difficult encounter situations different from the training scenarios, was also used as a test set to verify the effectiveness of the method in close-quarters situations in crowded waters.

The remainder of this paper is organized as follows: Section 3 introduces reinforcement learning. Section 4 describes the details of the algorithm design and includes the design of the observation state, action space and reward function, as well as the arrangement of training scenarios. In Section 5, we carry out validation and a case study. Section 6 is the conclusion and prospect.

3. Reinforcement Learning

Reinforcement learning is a goal-oriented learning method that can understand and automate decision-making problems. It emphasizes that agents learn through direct interaction with the environment without imitable supervision signals or complete modeling of the surrounding environment. Therefore, it has a different paradigm from other computing methods. The agent in reinforcement learning maps the current situation into actions to maximize the reward. The agent is not told which action should be taken but must discover for itself by trying to find out which action will yield the greatest benefit. Actions in the decision-making process affect not only the immediate gain but also the next scenario and thus the future return. Trial and error and delayed gain are two of the most significant

characteristics of reinforcement learning. In addition to agent and environment, there are four core elements of a reinforcement learning system: policy, reward signal, value function and (unnecessary) modeling of environment [27].

Policy defines the behavior of agents in specific situations, that is, the mapping from environmental state to action. Reward signal defines the goal in reinforcement learning. The only purpose of the agent is to maximize long-term reward. Therefore, reward signal is the basis of changing the policy. The value function is different from the reward signal. The reward signal indicates the reward in a short time, whereas the value function indicates the reward from a long-term perspective. The choice of action is made based on the judgment of value, and it is also the most important component of a reinforcement learning algorithm. Environmental modeling is a simulation of a reactive model of the environment. For example, given a state and action, the model can predict the next state and next reward of the external environment. The architecture for the application of reinforcement learning algorithms to a ship collision avoidance system is shown in Figure 1. The reward signal in the collision avoidance system is the reward given to the agent by the environment, which is usually in the form of a function; the policy is the method of collision avoidance decision, and the choice of action is usually based on the strategy function. The DQN algorithm belongs to the off policy, so the action is determined by the action value function, which is fitted by a neural network; environment modelling is used to predict the future position, course and speed of the ship and the target ships by the ship motion model and ship position transfer probability model so as to calculate the observation state of the agent at the next moment. The specific process of DQN applied to ship collision avoidance should be as follows: the agent observes the state it is in now from the environment, calculates the value corresponding to all actions in this observation state by means of an action value function and randomly selects the action or chooses the action with the greatest value during the exploration phase. The random selection of actions is necessary not only to keep exploring future actions with greater payoffs but also to avoid getting stuck in a local optimum solution. The greedy strategy [28] (ε-greedy) is the most commonly used measure and selects the optimal action with probability p and chooses actions at random with a probability of 1-p. The agent will then perform this action, and the environment will update the state, as well as give the reward to the agent. At the same time, the agent stores the current moment's state, the action taken, the reward received and the next moment's state for use in learning to update the neural network parameters. The intelligence will continue to cycle in this pattern until it reaches its goal or fails.

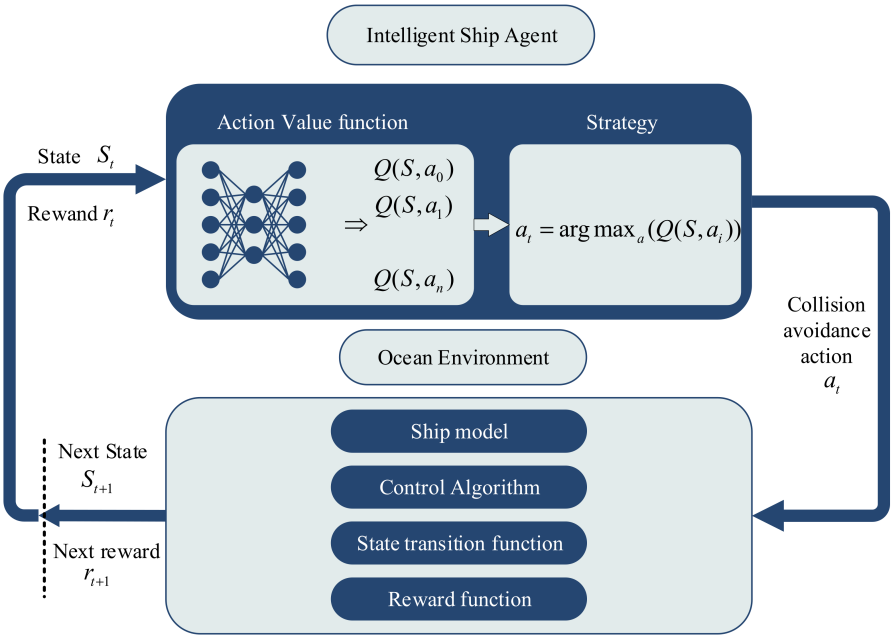


Figure 1. Configuration of DRL algorithm for collision avoidance.

4. RL Algorithm Design

In this section, we will discuss the design of observation state space, action space, neural network model, reward function and training scenarios in the RL algorithm. Figure 2 shows the flow chart of the algorithm. At the beginning of each episode, the scenario is first initialized and randomly selected from the scenario set. Then, distance of the closest point of approach (DCPA), time to the closest point of approach (TCPA), range, bearing and other parameters are calculated. If there is a risk of collision, the observation state will be input to the RL algorithm to generate a collision avoidance decision; if there is no danger of collision, the ship will be judged as to whether it has passed and cleared the target ship, and if not, the ship will continue to sail in the same course and at the same speed; if it has passed and cleared the target vessel, the ship will take an action to return to the planned route. The resulting action is fed to the rudder controller, which updates the ship's dynamic information in conjunction with the motion model. Then, the algorithm judges judge whether the end condition is satisfied: when the distances between the agent ship and target ships are less than the threshold (collision occurs) or the distance to the waypoint is less than the threshold and there is no risk of collision (collision avoidance success), end the round; otherwise continue to execute the cycle. Training ends when the training rounds reach the maximum.

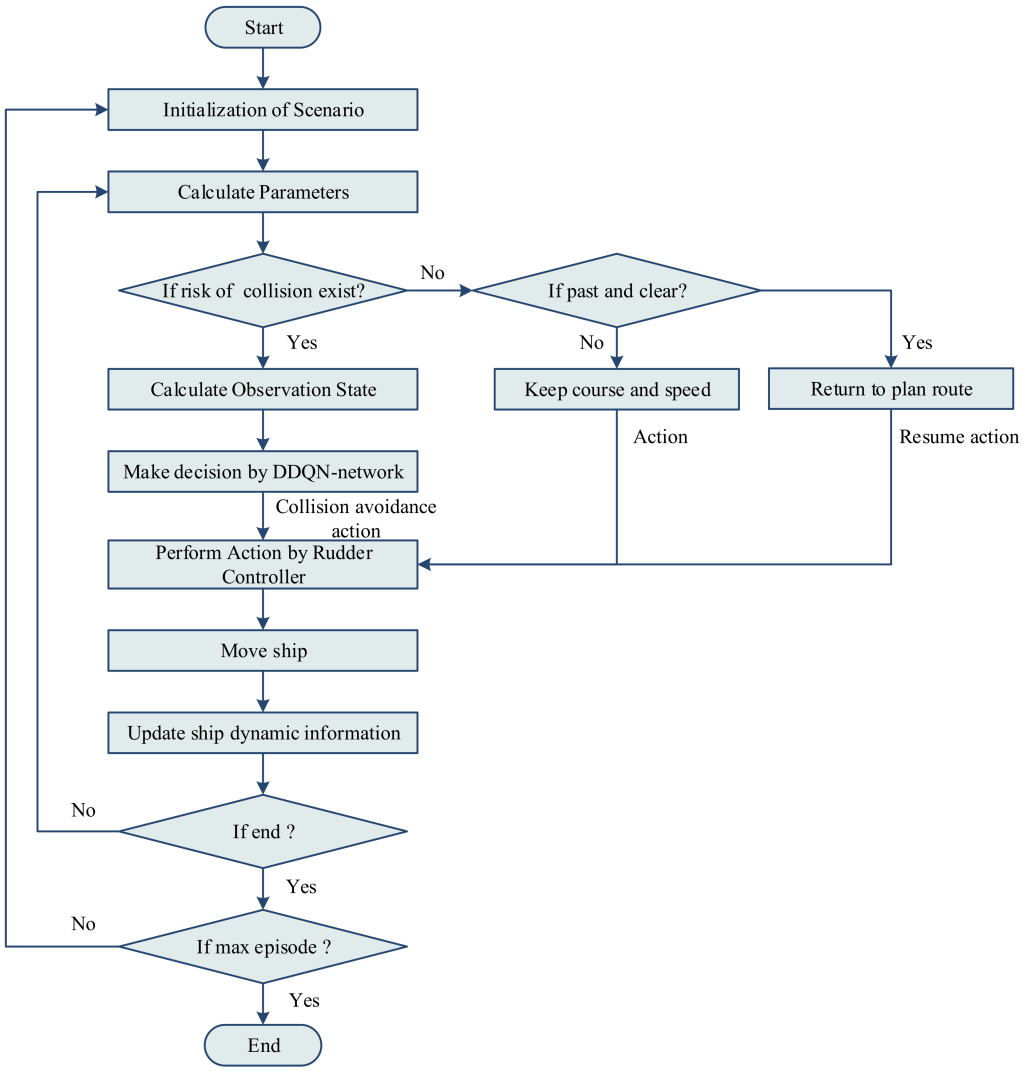


Figure 2. Flow chart of DRL algorithm for collision avoidance.

4.1. Observation State

RL algorithms rely heavily on the input of the observed state, and it can even be argued that the observed state directly influences the learning ability of the agent. In previous studies, environmental states, such as ship kinematic parameters, distance and bearing, have been used as inputs, but raw, unprocessed data are difficult to learn and often require a deeper and wider network structure to process. In this paper, we use an approach that quantifies the predicted hazardous areas where future collisions are likely to occur by the dynamic information received from external sensors about obstacles and target ships as the observation states of the DNN. This approach clusters several similar collision avoidance scenarios.

In past research, researchers have usually used the predict area of danger (PAD) model to predict the areas where ships may collide in the future. Junji Fukuto and Hayama Imazu proposed obstacle zone by target (OZT), a method based on risk evaluation circle, to predict the danger zone of collision between ships [29]. Figure 3a shows the PAD, and Figure 3b shows the OZT; however, neither the OZT nor the PAD takes into account the actual course of the ship. Based on the OZT and PAD, we simplify the prediction of dangerous areas, take the actual course of the agent ship into account and filter the ships without risk of collision, which can avoid the redundancy of plotting.

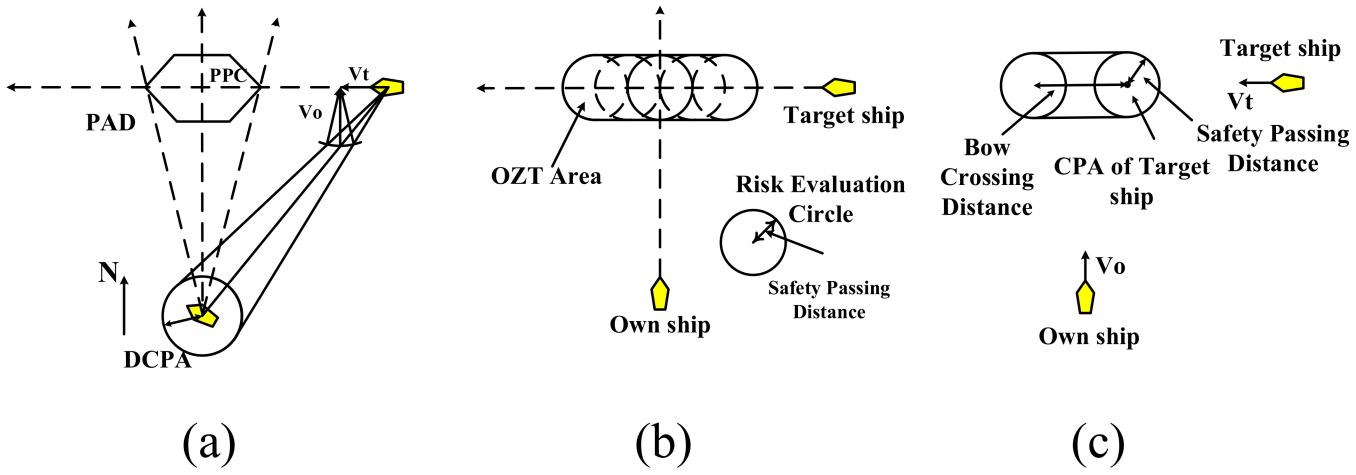


Figure 3. Schematic diagram of different methods of indicating potential collision area. (**a**–**c**) are PAD, OZT and predicted hazardous areas respectively.

Normally, the navigation status of ships involves course and speed, so the position of collision is usually near the closest point of approach (CPA) of the two ships. Therefore, the dangerous area shall be a circular area; the center of area is the CPA of the target ship, and the radius is the safety passing distance (SPD). It is the ordinary practice of seamen and good seamanship for seafarers to avoid crossing the bow of the other vessel when the two vessels meet. Thus, a longer distance should be reserved when crossing the other ship's bow to avoid invading the others ship domain. Thus, the predicted hazardous area shall be extended for bow crossing distance along the heading direction of the target ship. The agent ship shall avoid entering this area, as shown in Figure 3c. As the input of the neural network can only be a vector of fixed dimensionality, the algorithm should have the ability to process multiple ships at the same time. Therefore, in order to deal with this problem, we use the grid method to vectorize the predicted hazardous area to ensure that the observation vector can maintain the same dimension, regardless of the number of target ships. As shown in Figure 4, we adopt a concentric circle grid processing centered on the agent ship and extending outwards in all directions at fixed radius intervals and angular intervals. Take the ship's course as the starting point and clockwise as the positive direction to set the position. When the predicted hazardous area overlaps with the grid map, the component of the observation vector is set to 1; otherwise, it is set to 0. After repeated

attempts, the distance and angle interval of 0.5 NM and 10 degrees, respectively, are taken as the best values (Table 1). Therefore, the design input dimension of the algorithm is 1×433, with the 433rd component indicating that the collision risk area falls outside 6NM. This method can cluster similar scenes and also be used to display static obstructions. As shown in Figure 4, the red circle represents the agent ship, and the yellow, blue and brown circles and capsule-shaped areas represent the target ship and the predicted hazardous area, respectively. The orange polygons represent static obstructions. Figure 4 simulates a situation where the agent ship and three target ships meet at the same time. According to the COLREGs, the agent ship and the three target ships all constitute a pair head-on situation. Through this observation method, it can be found that the observation states between the agent ship and the three target ships are the same, which means that the agent can identify these three meeting situations as one, narrowing the observation state space.

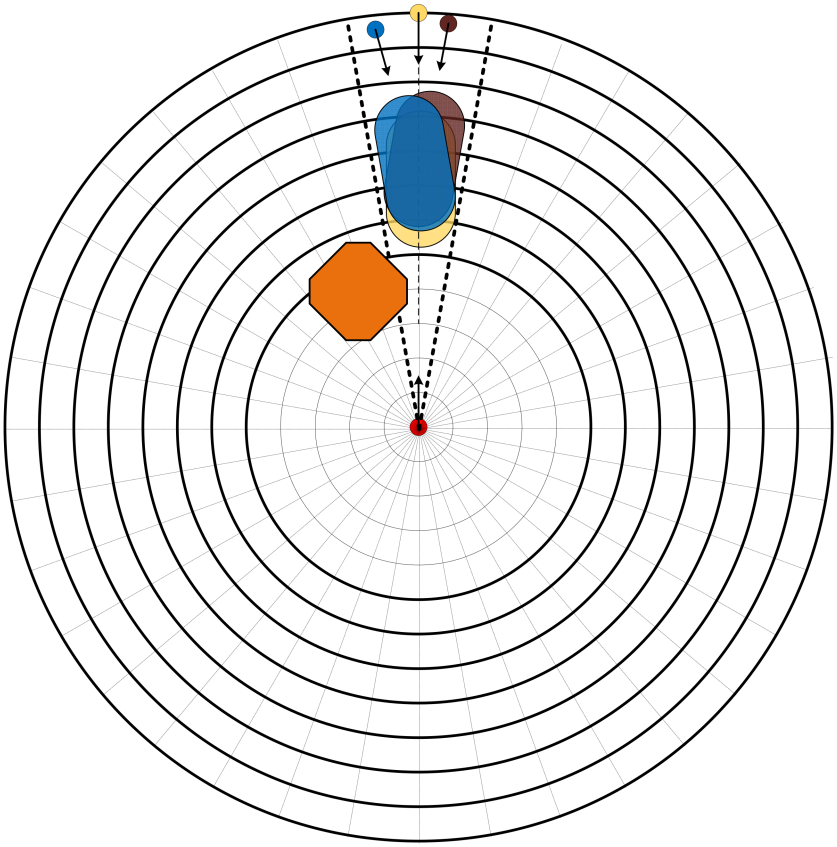


Figure 4. Observation state of agent.

Table 1. Configuration of grid.

Subject	Value
Angle range(°)	360
Angle interval(°)	10
Distance range(NM)	6
Distance interval(NM)	0.5
Bow crossing distance(NM)	1.0
Safety passing distance(NM)	0.5

4.2. Action Space

4.2.1. COLREGs

The importance of collision avoidance in navigational watch is recognized by the maritime community, as it often results in huge losses of life, cargo and ships, as well as serious pollution of the water environment. Furthermore, ship collisions often account for

the largest proportion of maritime accidents. Many of the requirements of the International Convention on Standards of Training, Certification and Watchkeeping for Seafarers (STCW) for the officer's navigational watch responsibility relate to collision avoidance at sea. In October 1972, the maritime departments of governments of various countries signed and adopted the 1972 Convention on the International Regulations for Preventing Collisions at Sea. After six amendments, the 1972 Convention on the International Regulations for Preventing Collisions at Sea became the current rules for preventing collisions. The rules clearly specify the applicable ships, waters and collision avoidance actions to be taken. The environment involved in a ship's autonomous collision avoidance system is mainly defined in Part B (steering and sailing rules). Rule 13 (overtaking), rule 14 (head-on situations) and rule 15 (crossing situations) stipulate the judgment of encounter situations between ships, the division of collision avoidance responsibilities and the collision avoidance actions to be taken. A vessel is deemed to be overtaking when coming up to another vessel from a direction more than 22.5 degrees abaft her beam. In the case of an overtaking situation, the giving-way vessel shall turn starboard or port to avoid other ships. When two vessels are meeting on reciprocal or near reciprocal courses so as to involve risk of collision, such a situation is deemed a head-on situation. In case of head-on situations, each ship shall alter her course to starboard so that each shall pass on the port side of the other. When two vessels are crossing so as to involve risk of collision, the vessel with the other on her own starboard side is the give-way vessel. In the crossing situation, the give-way vessel shall keep out of the way and alter her course to starboard to pass through the stern of the other ship if the circumstances of the case admit. In Figure 5, the red area represents the port crossing situation with other ships, the green area represents the starboard crossing situation, the pink area represents the head-on situation and the blue area represents the overtaking situation in which the agent ship is the overtaken vessel. Figure 6 shows the actions to be taken by each ship according to the COLREGs.

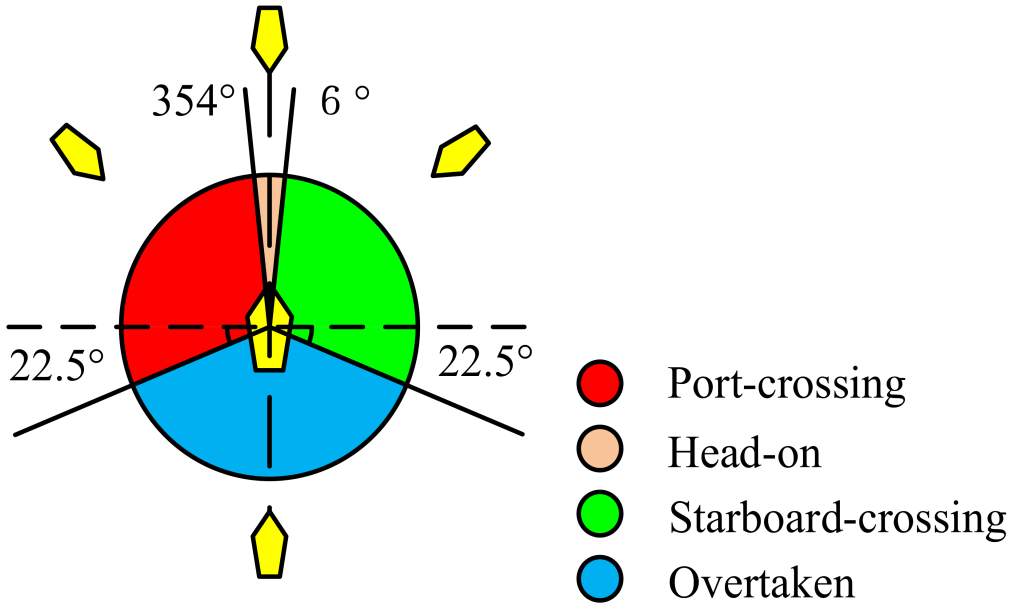


Figure 5. Encounter situations.

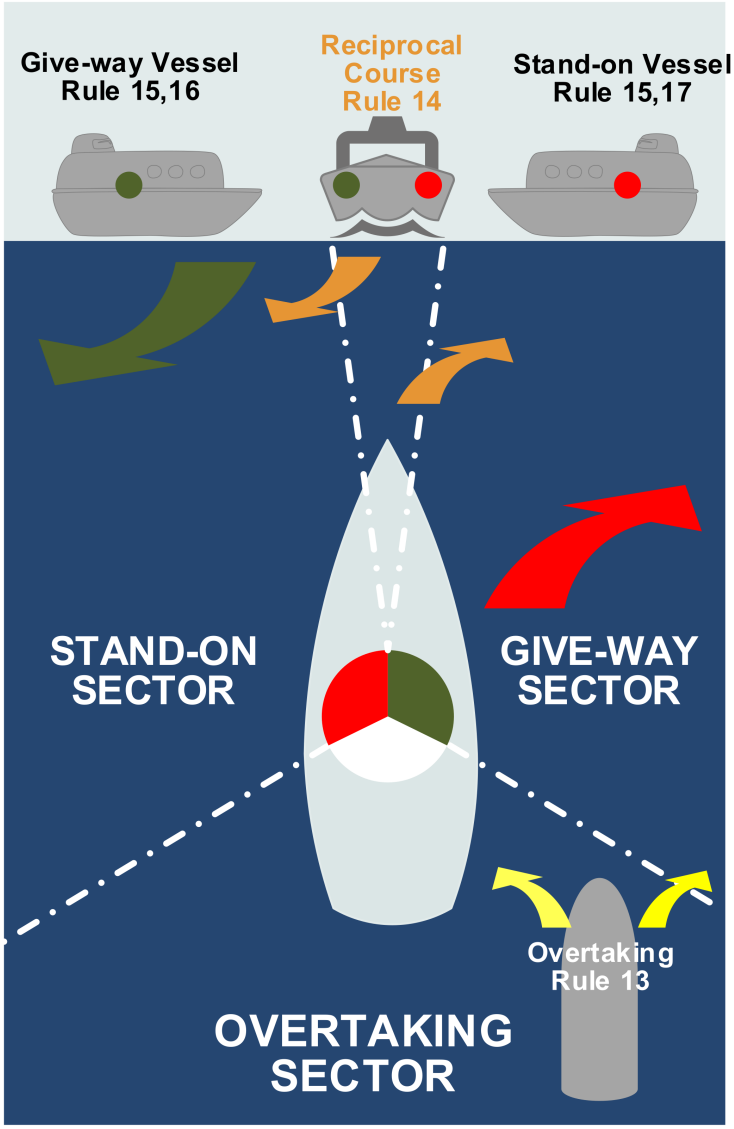


Figure 6. The actions to be taken by each ship according to the COLREGs.

4.2.2. Action Space Design

The ship collision avoidance process usually consists of four stages:

1. Environmental perception: detect the target ship; determine the encounter situation, whether there is a risk of collision and the time required to take collision avoidance action.
2. Take collision avoidance action: the give-way vessel shall take corresponding actions to keep clear other ships in accordance with COLREGs.
3. Keep course and speed: the give-way vessel shall drive in a straight line at a constant speed along the collision avoidance course until finally past and clear.
4. Return to the planned route: the ship shall return to the original planned route after finally past and clear.

In a complete collision avoidance process, the time occupied by keeping course, the speed stage and returning to the planned route is much longer than that required to take collision avoidance action, but the decision making is the key part of ship collision avoidance. Therefore, the application of an RL algorithm in the whole collision avoidance process will increase the complexity of the algorithm and cause the problem of difficult convergence of the model. In this paper, the collision avoidance algorithm is only used to take collision avoidance actions, which greatly improves the efficiency and robustness of the algorithm.

In the process of collision avoidance, the collision avoidance actions that can be taken by the OOW include steering course and changing speed. Due to the high inertia of the vessel, it is difficult to change speed, and the steering action is easily detected by the target vessel. The OOW therefore usually takes steering avoidance measures. In navigation practice, there are usually two kinds of decisions made by the pilot: one is to control the rudder angle, which turns the rudder or turns back at the appropriate position through empirical judgment to steer the ship to avoid collision; the other is to control the course of the ship an adjust the course to the appropriate course through a series of steering orders. As the first method depends on the maneuvering characteristics of ships, different ships adopt different rudder angles for collision avoidance under the same encounter, but the optimal steering angle is the same. Therefore, we take the second collision avoidance method as the action space, that is, a series of discrete steering orders to change course.

Usually, the six degrees of freedom (DOF) model is used to describe the motion of the ship, but in the field of ship collision avoidance the, 3 DOF model is usually used to describe ship motion. The coordinate system of the 3DOF model is shown in Figure 7. In Figure 7, v_x and v_y are the transverse component and longitudinal component of velocity, respectively; v and r are speed and head turning angular speed, respectively; ψ is the course of the ship, and δ is the rudder angle. To simulate maneuverability of the ship, we use the Nomoto equation [30] to calculate the ship motion, as expressed in Equation (1). In addition, the rudder angle is calculated by the PD controller (Equation (2)):

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} r \\ (K\delta - r)/T \\ (\delta_E - \delta)/T_E \end{bmatrix} \tag{1}$$

$$\delta_E = K_p(\psi_c - \psi) + K_d\dot{\psi} \tag{2}$$

where ψ is the course of the ship; ψ_c is the target course of the ship; r is the yow rate; K and T are the index parameters of ship maneuverability in clam water; δ and δ_E are the real rudder angle and command rudder angle, respectively; T_E is the time constant of the steering gear; and K_p and K_d are the controller gain coefficient and the controller differential coefficient, respectively.

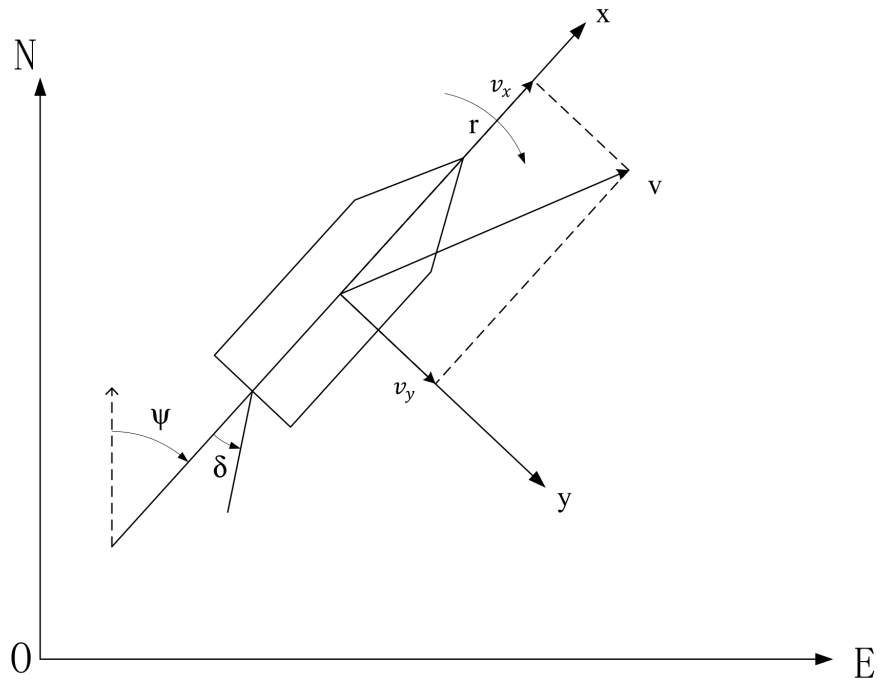


Figure 7. Coordinate system of ship motion.

According to the above contents and Rule 8 of the COLREGs, if there is sufficient sea-room, alterations of course alone may be the most effective action to avoid a close-quarters situation. The action space of the algorithm is discrete course change angle, the port steering is negative, and the starboard steering is positive, with an interval of 2° from $+12^\circ$ to -12°. The calculation formula of a ship's course is as follows in Equations (3) and (4):

$$\psi = \psi_{last} + a_t \tag{3}$$

$$a_t \in \{+12^\circ, +10^\circ, +8^\circ, +6^\circ, +4^\circ, +2^\circ, 0^\circ, -2^\circ, -4^\circ, -6^\circ, -8^\circ, -10^\circ, -12^\circ\} \tag{4}$$

where ψ_{last} is the course of the ship at the last sampling time, and a_t is the collision avoidance action.

4.3. Reward Function Design

The reward function is the core part of RL and directly affects the learning of the agent. In designing the reward function, we incorporated the COLREGs and expert experience to make the decisions made by the agent more like the results of human operation. Both rule 8 and rule 16 of the COLREGs stipulate collision avoidance actions. The action of the give-way vessel should be positive, made in ample time and with due regard to the observance of good seamanship. According to the recommendation of experts and scholars, collision avoidance usually starts when two ships are 5–8 NM apart. DCPA should be no less than 1 NM when 10,000-ton ships meet in daytime in good visibility, 1.5 NM in night or windy weather and greater than 2 NM in open waters in poor visibility.

In this paper:

1. When TCPA is less than 0, it is considered that there is no risk of collision (ROC);
2. When the distance from the target ship is greater than 2 NM and DCPA is less than 1.5 NM, there is considered to be a collision hazard;
3. When the distance is less than 2 NM and more than 1 NM and DCPA is less than 0.5 NM, a collision hazard exists.
4. When the distance is less than 1 NM and more than 0.5 NM and if DCPA is less than 0.3 NM, ROC exists.
5. Collision avoidance is considered to have failed when the distance between the two ships is less than 0.3 NM.
6. In the overtaking situation, when the distance between the two ships is less than 2 NM and the agent ship is a stand-on ship, the agent ship should take collision avoidance action alone to avoid immediate danger. In a crossing situation, where the distance from target ship is less than 4 NM, the ship should also act alone in order to reserve sufficient distance and time.

The reward function designed in this paper has four components, with a positive reward when the agent successfully avoids the target ship or static obstruction, i.e., when there is no ROC with any target ships (the observation state at the next moment is zero vector). This part of the reward takes the COLREGs and good seamanship into account. When the distance between the agent ship and any of the target ships is less than 0.3 NM, a collision is considered to have occurred, and a larger negative reward is given. When the ship moves into the predicted collision hazardous area, the agent will receive a smaller negative reward. In other cases, the reward is 0.

Five factors are considered when designing the reward for successful collision avoidance, namely the number of steering decisions, the amount of the cumulative steering angle, the deviation distance, the DCPA when clear of the other vessel and compliance with the COLREGs. Rule 8 states that, "Collision avoidance actions shall, be sufficient to allow easy observation by other vessels, as well as avoiding a series of minor changes to course speed". Therefore, intelligences should minimize the number of turns during collision avoidance and, in particular, should avoid swinging the bow from side to side. According to expert recommendations, the range of steering to avoid collisions should be at least 30 degrees. The effect of the avoidance action should be to clear the ship as

much as possible, so the DCPA between the two ships should be as large as possible. In order to meet the practical requirements, the ship should avoid collisions with as few unnecessary detours as possible to save fuel and time costs. Therefore, taking into account the requirements of the above rules and the actual situation, the reward function used in this algorithm is specified as follows.

$$reward = \begin{cases} [W_1 \ W_2 \ W_3 \ W_4 \ W_5 \ ] \left[R_{rudder} \ R_{\Delta\psi} \ R_{deviation} \ R_{clear} \ R_{COLREGs}\right]^T & if \ \forall s_{(t+1)_i} = 0 \ (i = 0, 1, \ldots, 432) \\ -10 & if \ d_i < 0.3 \ NM \\ -1 & if \ \forall s_{(t+1)_i} = 1 \ (i = 0, 1, \ldots, 35) \\ 0 & other \end{cases} \tag{5}$$

$$\begin{aligned} R_{rudder} &= (8 - n_{rudder}) \\ R_{\Delta\psi} &= \frac{|\Delta\psi|}{12} \\ R_{deviation} &= (2 - d_{deviation}) \\ R_{clear} &= \frac{1}{n_{ship}} \sum_{i}^{n_{ship}} DCPA_i \\ R_{COLREGs} &= r \end{aligned} \tag{6}$$

where n_{rudder} is the number of collision avoidance actions, $\Delta\psi$ is the course change angle and $d_{deviation}$ is the deviation distance. It is assumed that both the agent ship and target ships maintain course and speed, and the deviation distance is calculated by estimating the position of the agent ship when the requirements of returning to the planned route are met. In this paper, the requirements for pass and clear are that the agent ship is sailing along the resumed course without a renewed risk of collision with another ship. In particular, it should be noted that the course to be used for the calculation of the pass and clear should not be the current course of the agent ship but the resumed course towards the planned waypoint. n_{ship} is the number of target ships, and r is the part of rule reward. If the action complies with the COLREGs, r is +2; otherwise r is -2, s_{t+1} is the state at the next sample moment and d_i is the distance from the target ship. W_i is the weight of each part of reward, where $W_1 = 1$, $W_2 = 1$, $W_3 = 0.5$, $W_4 = 1$, $W_5 = 1.5$.

4.4. Scenario Set

In this paper, encounter situations are divided into six types: head-on situation, port crossing situation, starboard crossing situation, overtaking situation, overtaken situation and other situations. As shown in the Figure 8, 1 to 3 are head-on situations, 4 to 6 are overtaking situations, 7 to 11 are starboard crossing situations, 12 to 14 are overtaken situations and 15 to 19 are port crossing situations. When setting up the scenarios, the initial ship position, course and speed of each scenario are within a random range to ensure that the DCPAs with target ships are small enough. When designing the initial positions of target ships, they are 6NM away from the agent ship, which is in line with the ordinary practice of seafarers, whereby the OOW usually takes collision avoidance action when the target ship is 5–8 nautical miles away from the agent ship. In addition, the training set contains all the ship encounter situations described in the COLREGs and clusters the scenarios with similar initial observation states. The details of scenarios are shown in Table 2. The ship's position is set at (0,0), course is 000 and the destination waypoint is set at (0,13). During training, 19 encounter scenarios are combined through arrangement, combination and random selection, including both simple, single-ship encounter situations and complex, multi-ship encounter situations.

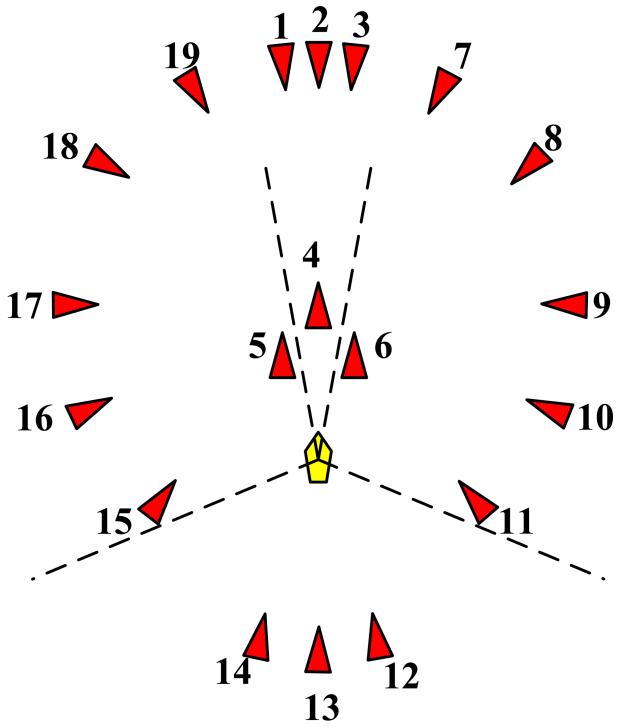


Figure 8. Scenario set of encounter situations.

Table 2. Scenario set.

Ship	X (NM)	Y (NM)	$\psi(^\circ)$	*Speed* (m/s)
Agent ship	0.00	0.00	000	10.00
Target Ship				
1	−0.63	5.96	178	10.00
2	0.00	6.00	180	10.00
3	0.63	5.96	182	10.00
4	0.00	3.00	000	4.00
5	−0.26	2.98	002	4.00
6	0.26	2.98	358	4.00
7	1.55	5.79	210	10.00
8	3.00	5.19	240	10.00
9	4.24	4.24	270	10.00
10	6.00	0.00	324	16.67
11	5.79	−1.55	315	18.73
12	0.26	−2.98	358	18.00
13	0.00	−3.00	000	18.00
14	−0.26	−2.98	002	18.00
15	−5.79	−1.55	045	18.73
16	−6.00	0.00	054	16.67
17	−4.24	4.24	090	9.24
18	−3.00	5.19	060	10.00
19	−1.55	5.79	030	10.00

4.5. DDQN with PER

V. Mnih et al. [28,31] proposed a deep Q network based on experience replay in 2013 and proposed the concept of target network in 2015, marking the birth of DQN. DQN is a mainstream and widely used deep reinforcement learning algorithm. In fact, once it was launched, it has a great impact on the field of reinforcement learning. When playing any Atari game, it can reach or even surpass the human level, only using the game screen as input. However, DQN often overestimates Q values with maximization deviation due to

the maximum operator and bootstrap. To solve this problem, the Deepmind team published a paper in 2015 proposing DDQN, which is solved by setting two independent Q networks and training each network independently [32]. One Q function is used to select actions, and the other Q function is used to evaluate actions. With experience replay, an agent transitions from one state, s, to the next state, s', by executing a given action, a, in the interaction with the environment and gets a reward, r. The transition information (s, a, r, s') is stored in an experience pool, and the algorithm learns from the experience pool. In DQN architecture, experience replay is used to ensure that the updates are uncorrelated. However, random or uniform sampling of transition information from playback memory is not the best approach. Instead, transition information can be selected and sampled according to priority. In the training process, the transition information with large temporal difference error (TD error), which is given higher priority, is sampled, which is conducive to the rapid and effective learning of the network [33].

In this paper, we use DDQN with a prioritized experience replay algorithm. During model training, this algorithm adopts two sets of neural networks with the same structure but different parameters, as well as temporary freezing correlation technology, to effectively solve the overestimation problem of natural DQN and uses prioritized experience replay (PER) [33] to reduce the amount of experience required for learning. The minibatch in the training set is given priority weight, which reduces the number of iterations in the training process and the training time of the model. The model is shown in Figure 9.

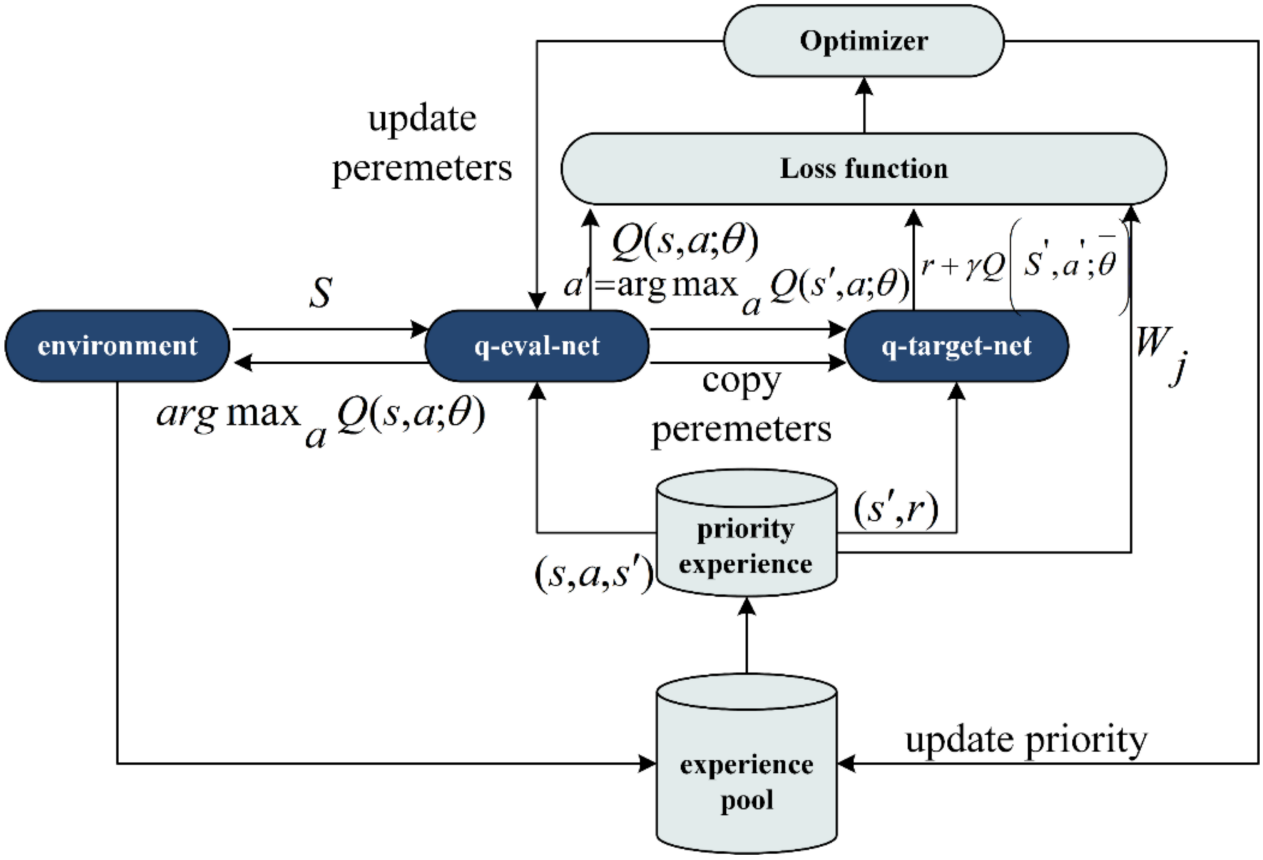


Figure 9. Structures of networks used in DDQN with prioritized experience replay.

Where θ is the parameter of q-eval-net, $\bar{\theta}$ is the parameter of q-target-net, a' is the next action and γ is the reward decay index.

First, the agent observes the environment state and selects the action through q-eval-net and returns it to the environment. Then, the environment returns the reward and the next state to the agent. The agent updates the state and stores (s, a, r, s') in the experience pool. Then comes the learning process: the agent first extracts the prioritized experience

from the pool and returns to the next action that can get the maximum reward in the next state by q-eval-net. Then, the next action and next state are input into q-target-net to obtain q-target. Then, the state is input into q-eval-net to get the q-eval of action and calculate the error between q-eval and q-target. The gradient descent method is used to update the parameters. Finally, the priority of experience is updated in the experience pool. After a fixed episode interval, the parameters of q-eval-net are copied to q-target-net.

The priority of experience and the probability of its selection are as follows:

$$p_i = |\delta_i| + \varepsilon \tag{7}$$

$$p_{(i)} = \frac{p_i^\alpha}{\sum_{i=1}^{k} p_i^\alpha} \tag{8}$$

where p_i is the priority of experience; δ_i is the TD error of experience; ε is a hyperparameter, which is a minimal number to prevent the experience with TD equal to 0 from being selected; $p_{(i)}$ is the probability of sampling experience being selected; α is the hyperparameter that controls the preference of sampling in uniform sampling and greedy sampling; and $\alpha \in (0, 1)$. When α is 0, the sampling is uniform; when it is 1, the sampling is greedy.

By defining priority and probability, we use Sumtree to select the experience that should be preferentially selected. Sumtree is a tree structure. Each leaf stores the priority, p_i, of each experience. Each branch node has only two forks. The value of the node is the sum of two forks. The top of Sumtree is the sum of all p_i. The batch size is k, the range, $[0, p_{total}]$, is divided into k ranges and then random numbers are generated in each interval and selected according to the corresponding strategies (van Hasselt et al., 2015).

With the use of prioritized experience replay, the distribution of samples is changed, which may cause the model to converge to different values. We use importance sampling, which ensures that each sample has a different probability of being selected and that they have the same effect on gradient descent. The importance sampling weight, w_j, is introduced into the loss function:

$$w_i = \left(\frac{1}{N} * \frac{1}{p_{(i)}}\right)^\beta \tag{9}$$

$$w_j = \frac{\left(N * p_{(j)}\right)^{-\beta}}{max_i(w_i)} \tag{10}$$

where w_i is the weight of the experience in the memory pool, w_j is the weight of the experience in the minibatch, N is the memory size, β is a hyperparameter used to offset the effect of prioritized experience replay on convergence results and $\beta \in (0, 1)$. When $\beta = 1$, the effect of prior experience is completely offset. Considering the priority of samples, the loss function is:

$$q_{eval} = Q(s_t, a_t; \theta_t) \tag{11}$$

$$q_{target} = r + \gamma Q\left(s_{t+1}, argmax_a Q(s_{t+1}, a; \theta_t); \theta_t^-\right) \tag{12}$$

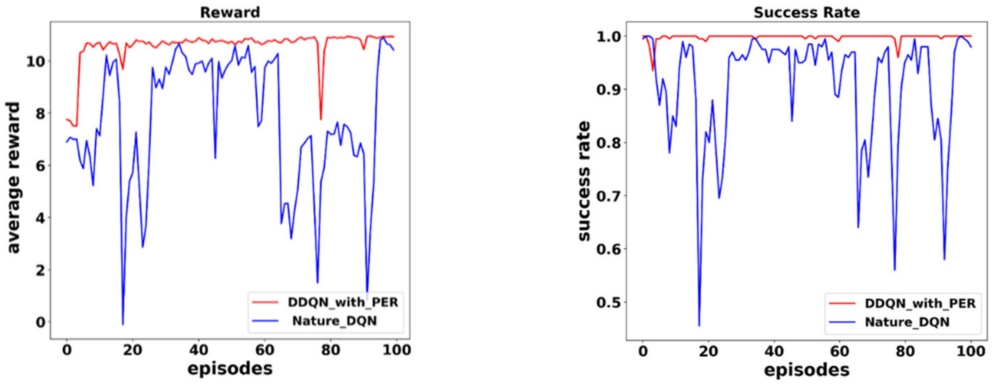
$$loss\ function = \frac{1}{k} \sum_{j=1}^{k} w_j \left(q_{eval} - q_{target}\right)^2 \tag{13}$$

We used Pytorch to build the neural network, Tkinter to realize the visual display and Matplotlib to draw figures. The network consists of five layers. The first layer is the input layer, with 433 nodes using the LeakyRelu activation function. Three hidden layers, 512 nodes in the first hidden layer, 256 nodes in the second hidden layer and 128 nodes in the third hidden layer, use the Relu activation function. There are 13 nodes in the output layer using the Softmax activation function. See for details of hyperparameters in Table 3.

Table 3. Hyperparameters for DDQN with PER.

Parameter	Value
DDQN	
Learning rate	0.005
Reward decay γ	0.95
ε-greedy	0.9
Replace target-net step	500
Memory size N	5096
Bitch size k	512
Loss function	MSE
Optimizer	Adam
Input layer nodes	433
Activation function	LeakyReLU (0.2)
Hidden layer, one node	512
Activation function	ReLU
Hidden layer, two nodes	256
Activation function	ReLU
Hidden layer, three nodes	128
Activation function	ReLU
Output layer nodes	13
Activation function	SoftMax
PER	
α	0.4
β	0.6

Before training, we compare the performance of DDQN with the PER algorithm with that of Nature DQN. Both algorithms adopt the same hyperparameters and the same single-ship encounter scenario (scenario 2 mentioned above), and a total of 20,000 episodes are trained; the ε initial value is 0.9, and every 1000 episodes, ε increases by 0.05. The q-eval-net parameters are copied to q-target-net every 200 episodes. The network is deemed to have converged when rewards received by the agent are almost unchanged and the agent has sailed without collision. Figure 10a shows the average reward obtained by the agent every 200 episodes; the red line represents DDQN with PER, and the blue line represents Nature DQN. Figure 10b shows the success rate of agent collision avoidance. The results show that compared with Nature DQN, DDQN with PER can converge rapidly and obtain greater rewards by learning the experience with larger TD error. In terms of success rate, DDQN achieves a success rate of about 99% after a few episodes.



(**a**) (**b**)

Figure 10. Comparison of average reward and success rate of DDQN with PER/Nature DQN. (**a**) is the average reward obtained by the agent every 200 episodes and (**b**) is the success rate of agent collision avoidance.

5. Simulation and Discussion

In this section, we discuss the performance of the algorithm on the training set and the test set. In the training set, we illustrate two typical single-ship encounter scenarios and an extremely difficult multi-ship encounter scenario with three target ships. To prove the effectiveness of the algorithm, we use the Imazu problem, which is regarded as a series of difficult ship encounter scenarios. In setting up its initial scenario, we set the distance between ships closer to each other, even forming a close-quarters situation. These scenarios simulate a situation where the target ship is already very close to an intelligent ship when detected by the perception system. These scenarios are completely different from those in the training set and are used to test the performance of the model in a completely unknown and urgent situation. In the following figures, the red circle represents the current position of the agent ship, and target ships and the predicted hazardous area are represented by circles and capsules of different colors. The display mode in the figure is relative motion display, and the course of the agent ship is upward. In the figures, the spacing between each concentric circle is 0.5 NM, and the spacing between each azimuth line is 10 degrees. In the simulation experiments, it is assumed that the target vessels maintain their course and speed, although they are also responsible for avoiding. In addition, the waters simulated in this model are open water. The model used in the simulation experiment is the teaching ship of Dalian Maritime University "YuKun", with length between perpendiculars (L), breadth (B), turning ability index (K) and following index (T) of 105 m, 18 m, 0.34 1/s and 64.74 s, respectively [34].

5.1. Simulation in Training Set

5.1.1. Case 1: Head-On Situation

The initial information of the agent ship and target ships is shown in Table 4. The coordinates take the agent ship as the center; due east and due north are the positive directions of the X axis and Y axis, respectively; the course is expressed by the circumference method; and the speed unit is knots.

Table 4. Head-on situation.

Ship	X (NM)	Y (NM)	ψ $(^\circ)$	$Speed$ (kn)
Agent ship	0.00	0.00	000	10.00
Target Ship	0.00	6.00	180	10.00

The experimental results are shown in the Figure 11. Figure 11a shows the observation state of the ship at the initial moment. The agent ship takes collision avoidance action one, $a1 = +12^\circ$, turning to 012. The state transition after the control system has executed the order is shown in Figure 11b. The agent ship's course is 012 in order to avoid the risk of collision with the target ship. Then agent ship then takes collision avoidance action two, $a2 = +12^\circ$. The observation status after $a2$ is executed is shown in Figure 11c, and the next action taken is $a3 = +10^\circ$. After the execution, the ship goes to heading 034, and there is no risk of collision with the target ship, as shown in Figure 11d, the agent ship shifts mode to maintain course and speed. At this time, the ship will sail along the heading for a period until the return requirements are met. Finally, the ship will sail to the destination waypoint to end the collision avoidance process. According to the COLREGs, the agent ship should turn to starboard to avoid the target ship. The experimental results show that the actions taken by the ship are in line with the rules of collision avoidance to let other ships clear. The ship's trajectory and the distance between the two ships at different times are shown in Figure 12.

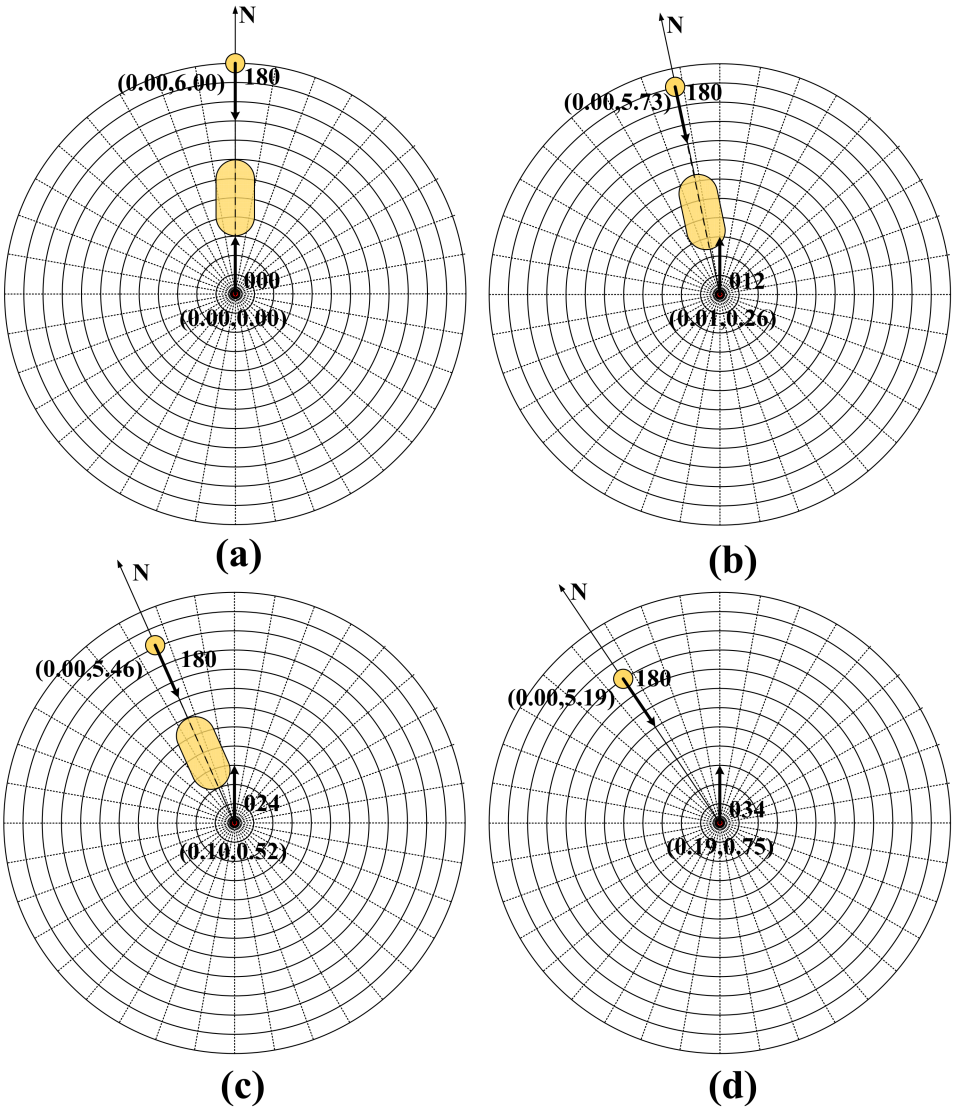


Figure 11. The observation state transition of collision avoidance in head-on situation. (**a**–**d**) shows the collision avoidance process.

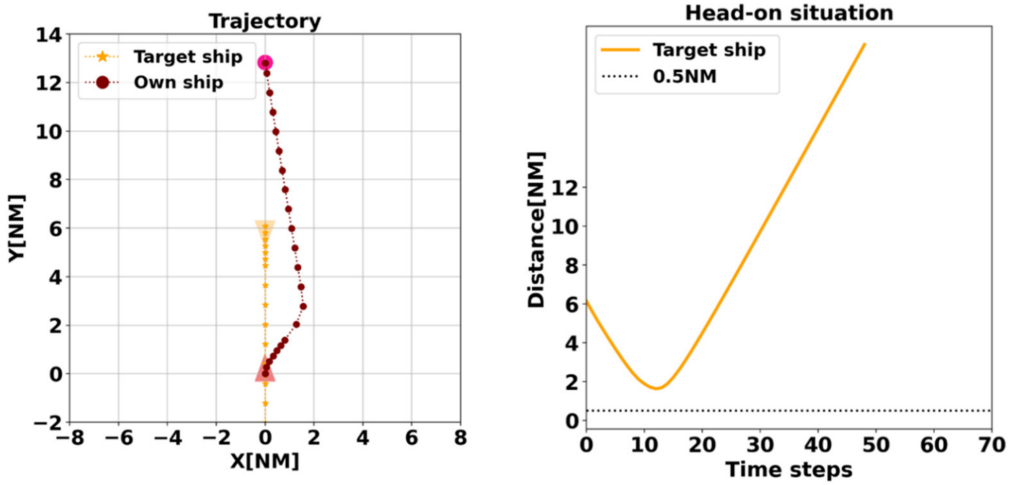


Figure 12. Trajectory of the agent ship and target ship and distance between ships in head-on situation.

5.1.2. Case 2: Crossing Situation

The simulation results are shown in the Figure 13. Similarly to the head-on situation, the agent ship avoids the target ship after taking turning actions. The initial information of the agent ship and the target ship is shown in Table 5. Figure 13a–d describes the state transition process of collision avoidance, and Figure 14 shows the ship trajectory and the distance between the two ships.

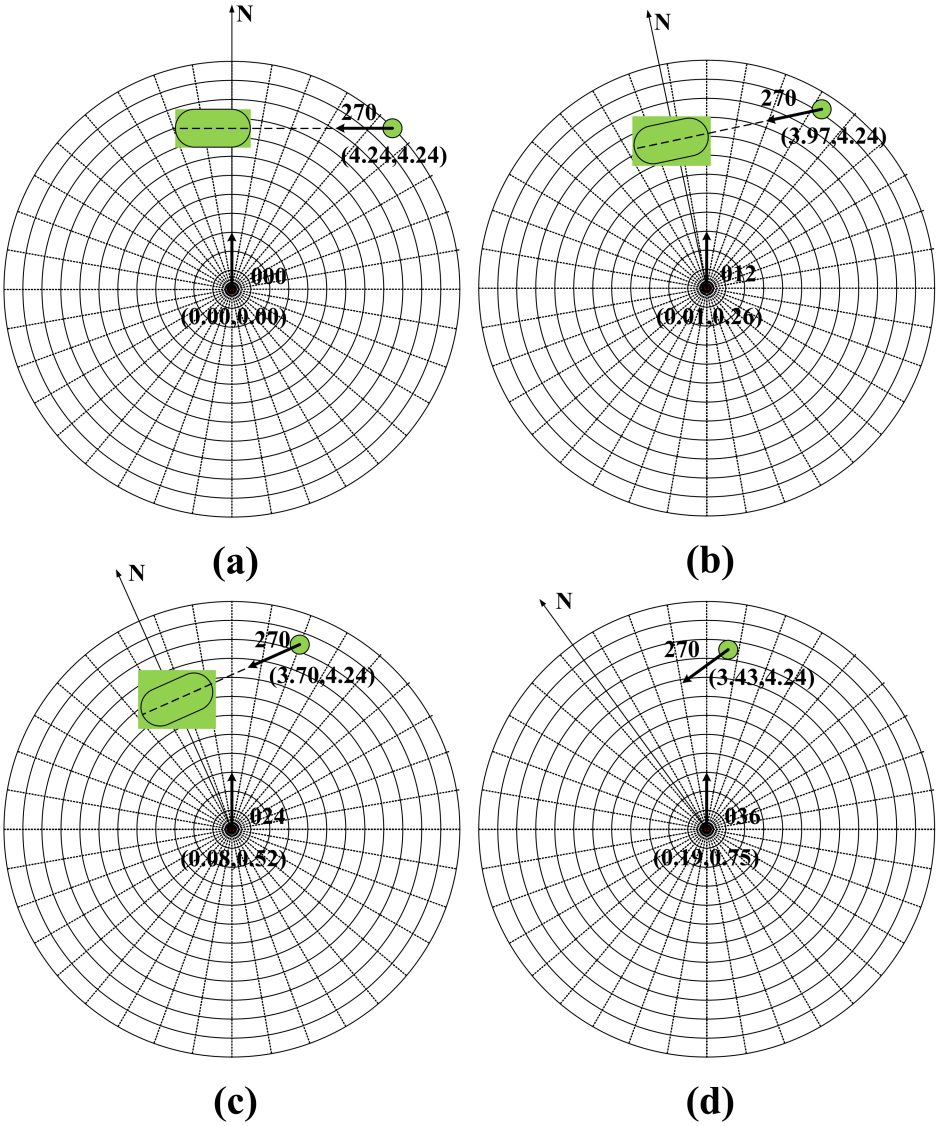


Figure 13. The observation state transition of collision avoidance in crossing situation. (**a**–**d**) shows the collision avoidance process.

Table 5. Crossing situation.

Ship	X (NM)	Y (NM)	ψ $(^\circ)$	$Speed$ (kn)
Agent ship	0.00	0.00	000	10.00
Target Ship	4.24	4.24	270	10.00

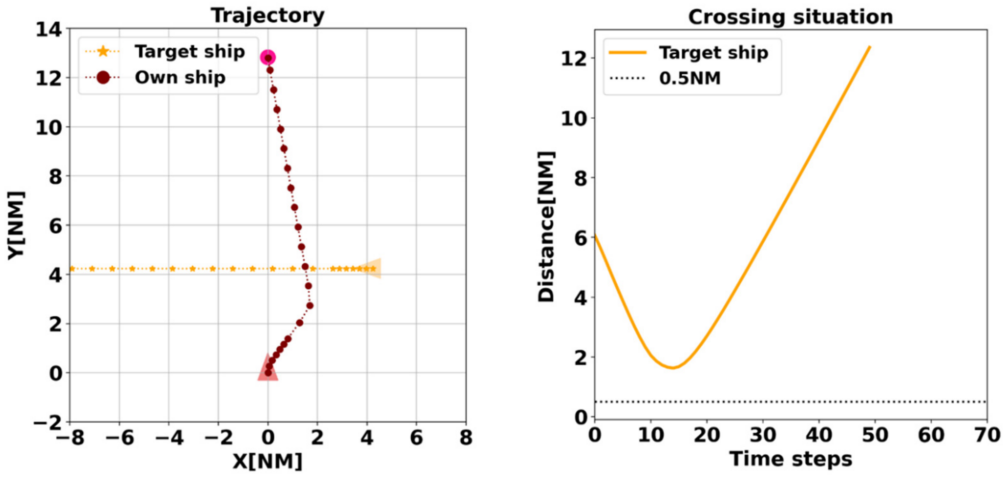


Figure 14. Trajectory of the agent ship and target ship and distance between ships in crossing situation.

5.1.3. Case 3: Multi-Ship Encounter Situation

The initial motion information of the agent ship and the target ships (TSs) is shown in Table 6. The experimental results are shown in Figure 15. The yellow, blue and green circles in the figure represent the positions of TS1, TS2 and TS3, and the capsule area of the corresponding color represents the predicted hazardous area. According to the COLREGs, the agent ship and TS1 form a crossing situation, and the agent ship is the give-way vessel; it forms a left crossing situation with TS2, and TS2 is the give-way vessel; the encounter situation with TS3 is a head-on situation, and the agent ship and TS3 shall alter course to starboard side together to keep out of the way. The state transition of the collision avoidance process is shown in Figure 15a–i. The ship's final course is 096. The agent ship successfully keeps clear of other ships. The ship's trajectory and the distance between the agent ship and the target ships are shown in Figure 16. Although the COLREGs do not delineate the situation of a multi-vessel encounter and do not specify the collision avoidance actions to be taken, the rules still play the role of legal and technical regulation. Therefore, taking collision avoidance actions between two ships in compliance with the COLREGs is considered good seamanship. In a realistic collision avoidance scenario at sea, ships should communicate with each other, understand each other's intentions and coordinate their collision avoidance actions. However, the purpose of the test is to verify the performance of the algorithm; the model should be such that even if the target ship takes no action or an uncoordinated action, the agent ship is still in a safe position and sails through at a safe passing distance.

Table 6. Multi-ship situation.

Ship	X (NM)	Y (NM)	ψ $(^\circ)$	*Speed* (kn)
Agent ship	0.00	0.00	000	10.00
Target Ship				
1	5.79	−1.55	270	18.73
2	−6.00	0.00	054	16.67
3	0.00	3.00	000	4.00

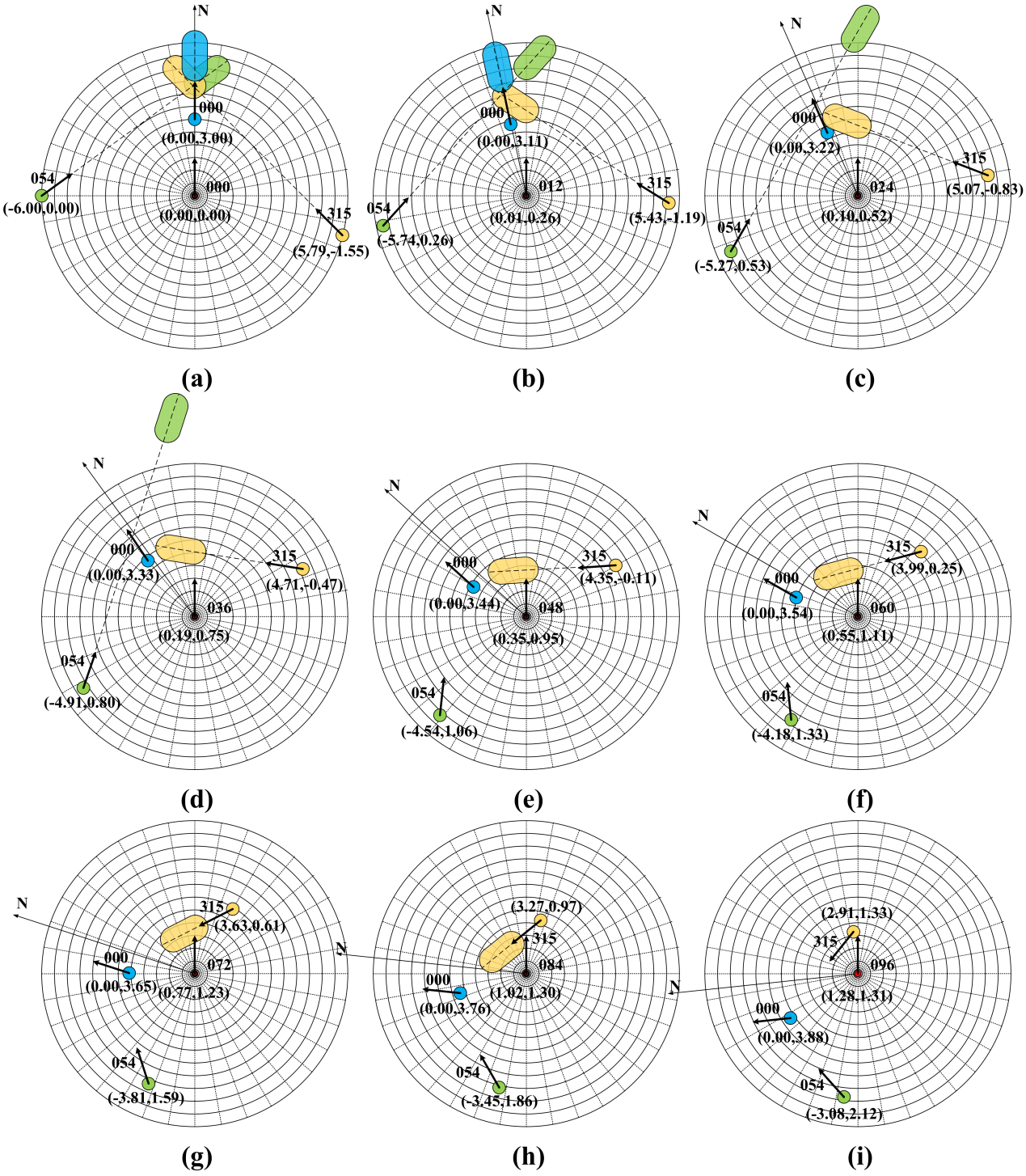


Figure 15. The observation state transition of collision avoidance in multi-ship encounter situation. (**a**–**i**) shows the collision avoidance process.

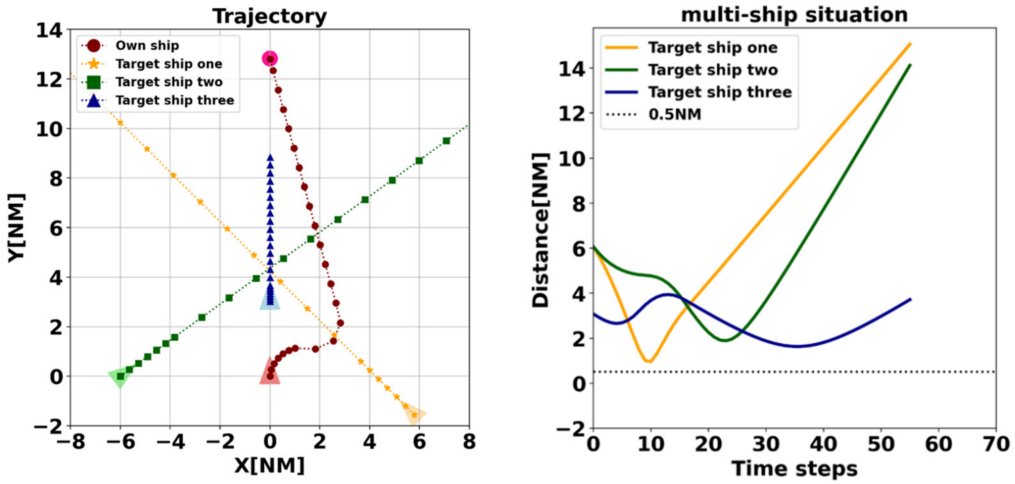


Figure 16. Trajectory of the agent ship and target ships and distances between ships in multi-ship encounter situation.

5.2. Simulation in Test Set

The method is simulated and tested in a complex environment different from the training set. We use the Imazu problem for verification. The Imazu problem is considered a series of extremely difficult ship encounter situations (Cai et al., 2013), of which 1–4 are single-ship encounter situations, 5–11 involve two target ships, and 12–22 involve three target ships. The initial position of the ship is (0.00,0.00), the course is 000 and the speed is 10 kn. The initial position and course of the target ships are shown in Table 7. The speed of target ships is 10 kn, and the speed of the overtaken target ships is 4 kn. It is assumed that the target ships maintain course and speed, and the position of the destination waypoint is (0.00,12.82). In some scenarios, such as TS2 in scenario 6, the initial position is already less than the safe passing distance from the agent ship during training, forming a close-quarters situation with the agent ship. The simulation results are shown in the Figures 17 and 18. Figure 17 shows the trajectory diagram corresponding to own ship and target ships, which are respectively represented by different marks of different colors; Figure 18 shows the distance between own ship and target ships, which is represented by solid lines of different colors. The dotted line in the figure represents 0.5 NM line, which is the safe passing distance in close-quarter situation. The simulation results show that in scenarios 1–3, the agent is able to take collision avoidance action well in accordance with the rules. In case4, own ship, as a stand-on ship, can still successfully avoid and sail through more than 0.5 nautical miles away in a close-quarters situation where the giving way ship does not act in time. In scenarios with two target ships, if there is a situation where the target ship forms a right cross with the ship, it has a greater influence on own ship's decision. The results show that the ship will turn to the right to avoid other ships, due to the influence of the rule compliance component of the reward function. In the scenario with three target ships, own ship makes a similar decision to the two-vessel scenarios. In case19 and case21, the ship makes the same collision avoidance decision, due to the influence of a target ship with a large angle crossing on the starboard side in both cases. Overall, the algorithm can cope with multi-vessel encounters in crowded open water, even in close-quarters situations. By analyzing its collision avoidance trajectory, own ship's behavior is consistent with the COLREGs. There is no crossing of the bow of the target vessel and several small-angle turns. There was also no turning to the left towards the port side of the vessel at close range. In these 22 cases, the distance between own ship and other ships at any time is greater than 0.5 nautical miles and does not cross ahead of other ships.

Table 7. Cases of Imazu problem.

Ship	**Target Ship One**			**Target Ship Two**			**Target Ship Three**		
Case No.	X (NM)	Y (NM)	ψ $(^\circ)$	X (NM)	Y (NM)	ψ $(^\circ)$	X (NM)	Y (NM)	ψ $(^\circ)$
1	3.00	6.00	180	-	-	-	-	-	-
2	3.00	3.00	270	-	-	-	-	-	-
3	0.00	3.00	000	-	-	-	-	-	-
4	−2.12	0.88	045	-	-	-	-	-	-
5	0.00	6.00	180	3.00	3.00	270	-	-	-
6	2.12	0.88	315	1.04	0.14	345	-	-	-
7	2.12	0.88	315	0.00	3.00	000	-	-	-
8	0.00	6.00	180	3.00	3.00	270	-	-	-
9	3.00	3.00	270	1.50	0.40	330	-	-	-
10	3.00	3.00	270	−1.04	0.14	015	-	-	-
11	1.50	0.40	330	−3.00	3.00	090	-	-	-
12	0.00	6.00	180	2.12	0.88	315	−0.78	0.10	015
13	0.00	6.00	180	−1.04	0.14	015	−2.12	0.88	045
14	3.00	3.00	270	2.12	0.88	315	−0.78	0.10	015
15	3.00	3.00	270	2.12	0.88	315	0.00	3.00	000
16	3.00	3.00	270	−2.12	0.88	045	−3.00	3.00	090
17	2.12	0.88	315	−1.04	0.14	015	0.00	3.00	000
18	2.12	5.12	225	1.50	0.40	330	1.04	0.14	345
19	2.12	5.12	225	1.04	0.14	345	−1.04	0.14	015
20	3.00	3.00	270	1.04	0.14	345	0.00	3.00	000
21	3.00	3.00	270	1.04	0.14	345	−1.04	0.14	015
22	3.00	3.00	270	1.50	0.40	330	0.00	3.00	000

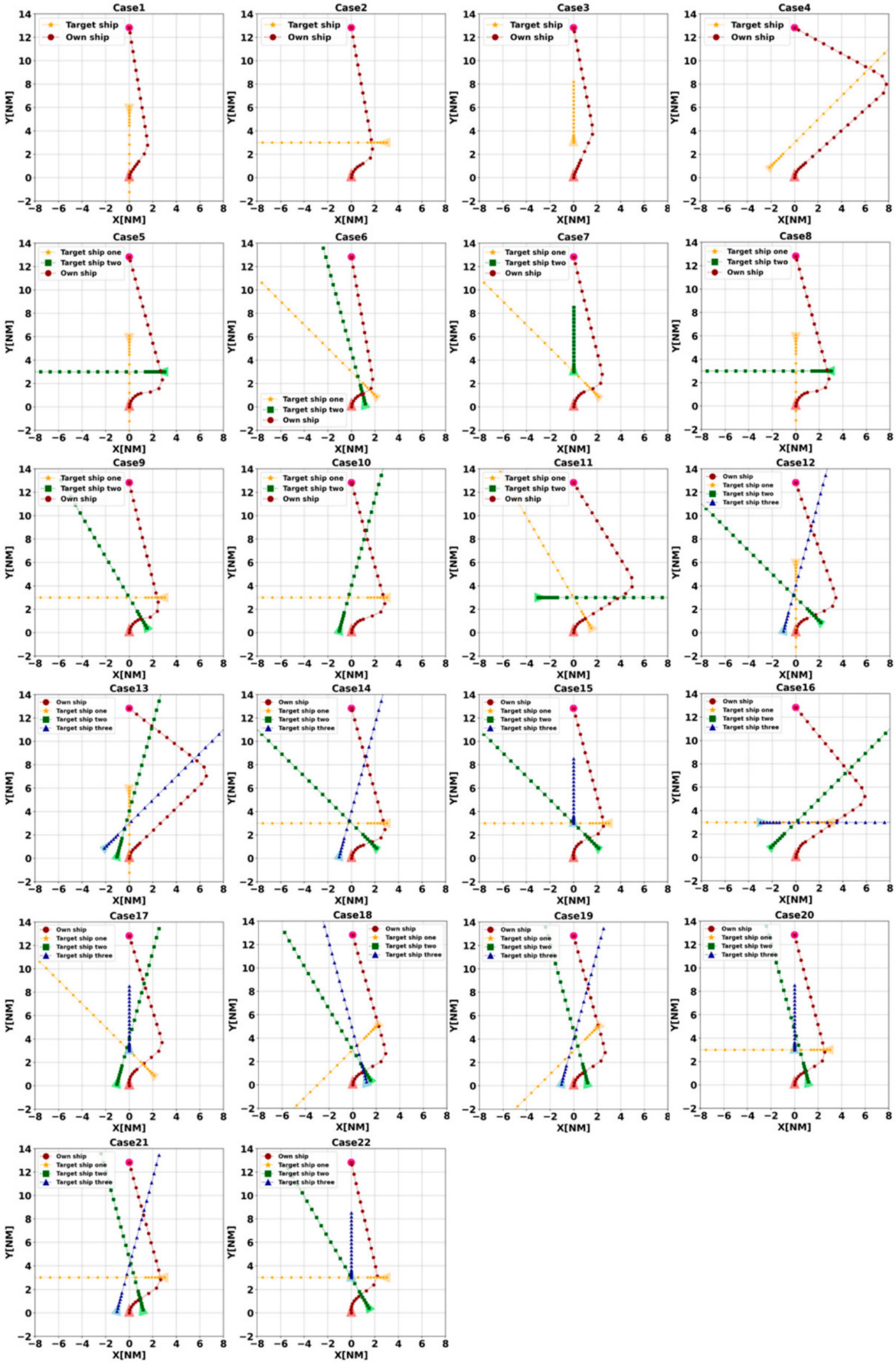


Figure 17. Trajectories of the agent ship and target ships in Imazu problem cases.

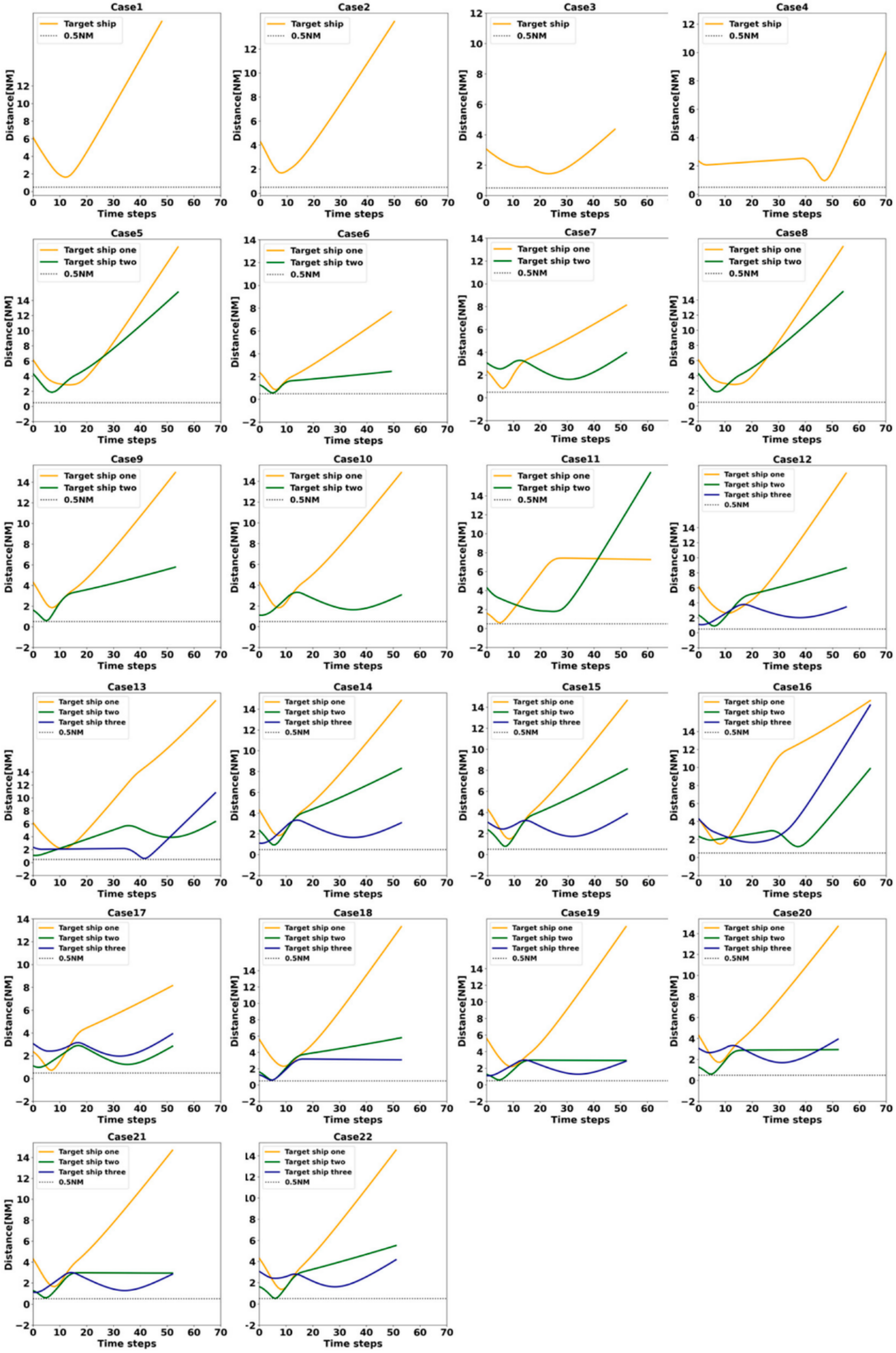


Figure 18. Distances between the agent ship and target ships in Imazu problem cases.

6. Conclusions and Future Work

In this paper, a multi-ship automatic collision avoidance method based on DDQN architecture is proposed. We vectorize the predicted hazardous areas as the observation states of the agent so that similar ship encounter scenarios can be clustered and the model can cope with any number of target ships. In order to achieve decision-making of the agent close to that of the human level, we designed a reward function based on the COLREGs and human experience, taking into account the five main factors considered by the OOW in a real collision avoidance situation. To speed up the learning process, DDQN with Prioritized experience replay is used to improve natural DQN. Before training the model, we compare the performance of the traditional DQN with the improved DQN using the same scenario and the same neural network model parameters, and the results show a significant improvement in the learning ability of the agent. Finally, 19 single-vessel collision avoidance scenarios are constructed based on the encounter situations classified by the COLREGs, which are arranged and combined as the training set of the agent. The Imazu problem is used to validate the effectiveness of the collision avoidance algorithm in close-quarters situations. The test results show that the algorithm can cope with multi-vessel encounter situations in crowded open water, even in close-quarters situations. The decisions made by the agent are in line with the COLREGs and close to human-level.

There are shortcomings of this study. For example, in case 4 of the Imazu problem, in a realistic situation at sea, sometimes the OOW may take a greater angle to the right than the agent's decision to avoid and, after sailing for a sufficient distance, would turn sharply to the left to pass the stern of the target ship to markedly reduce the deviation distance and time. The application of DRL in the field of ship collision avoidance is still in the exploratory stage, and there is still a gap between DRL and realistic collision avoidance at sea. In future research, we will build on this algorithm to design ship navigation, collision avoidance and control algorithms for restricted waters. In realistic multi-vessel collision avoidance scenarios at sea, collision avoidance actions are coordinated through communication, so we will try to use the multi-agent deep reinforcement learning (MADRL) distributed coordination method to implement multi-vessel collision avoidance.

Author Contributions: Conceptualization, P.Z. and Y.Z.; methodology, P.Z.; software, P.Z.; validation, P.Z., Y.Z. and W.S.; formal analysis, P.Z.; investigation, Y.Z.; resources, Y.Z.; data curation, P.Z. and W.S.; writing—original draft preparation, P.Z.; writing—review and editing, P.Z., Y.Z. and W.S.; visualization, P.Z. and W.S.; supervision, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Liao Ning Revitalization Talents Program (No. XLYC1902071), the National Key R&D Program of China (No. 2018YFB1601502), the Fundamental Research Funds for the Central Universities (No. 3132019313).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Statheros, T.; Howells, G.; Maier, K.M. Autonomous Ship Collision Avoidance Navigation Concepts, Technologies and Techniques. *J. Navig.* **2007**, *61*, 129–142. [CrossRef]
2. Ugurlu, H.; Cicek, I. Analysis and assessment of ship collision accidents using Fault Tree and Multiple Correspondence Analysis. *Ocean Eng.* **2022**, *245*, 110514. [CrossRef]
3. Tan, G.; Zou, J.; Zhuang, J.; Wan, L.; Sun, H.; Sun, Z. Fast marching square method based intelligent navigation of the unmanned surface vehicle swarm in restricted waters. *Appl. Ocean. Res.* **2020**, *95*, 102018. [CrossRef]
4. Miyoshi, T.; Fujimoto, S.; Rooks, M.; Konishi, T.; Suzuki, R. Rules required for operating maritime autonomous surface ships from the viewpoint of seafarers. *J. Navig.* **2022**, *75*, 1–16. [CrossRef]
5. Lyu, H.; Yin, Y. Fast Path Planning for Autonomous Ships in Restricted Waters. *Appl. Sci.* **2018**, *8*, 2592. [CrossRef]
6. Shaobo, W.; Yingjun, Z.; Lianbo, L. A collision avoidance decision-making system for autonomous ship based on modified velocity obstacle method. *Ocean Eng.* **2020**, *215*, 107910. [CrossRef]
7. Huang, Y.; Chen, L.; Van Gelder, P.H.A.J.M. Generalized velocity obstacle algorithm for preventing ship collisions at sea. *Ocean Eng.* **2019**, *173*, 142–156. [CrossRef]

8. Liu, C.; Mao, Q.; Chu, X.; Xie, S. An Improved A-Star Algorithm Considering Water Current, Traffic Separation and Berthing for Vessel Path Planning. *Appl. Sci.* **2019**, *9*, 1057. [CrossRef]
9. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D.; Khan, A. A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. *Ocean Eng.* **2018**, *169*, 187–201. [CrossRef]
10. Ni, S.; Liu, Z.; Cai, Y. Ship Manoeuvrability-Based Simulation for Ship Navigation in Collision Situations. *J. Mar. Sci. Eng.* **2019**, *7*, 90. [CrossRef]
11. Mohamed-Seghir, M.; Kula, K.; Kouzou, A. Artificial Intelligence-Based Methods for Decision Support to Avoid Collisions at Sea. *Electronics* **2021**, *10*, 2360. [CrossRef]
12. Borkowski, P.; Pietrzykowski, Z.; Magaj, J. The Algorithm of Determining an Anti-Collision Manoeuvre Trajectory Based on the Interpolation of Ship's State Vector. *Sensors* **2021**, *21*, 5332. [CrossRef] [PubMed]
13. Namgung, H.; Jeong, J.S.; Kim, J.-S. Real-Time Collision Risk Assessment System Based on the Fuzzy Theory in Accordance with Collision Avoidance Rules. In Proceedings of the 2020 Joint 11th International Conference on Soft Computing and Intelligent Systems and 21st International Symposium on Advanced Intelligent Systems (SCIS-ISIS), Hachijo Island, Japan, 5–8 December 2020; pp. 1–4.
14. Xie, S.; Garofano, V.; Chu, X.; Negenborn, R.R. Model predictive ship collision avoidance based on Q-learning beetle swarm antenna search and neural networks. *Ocean Eng.* **2019**, *193*, 106609. [CrossRef]
15. Li, J.X.; Wang, H.B.; Zhao, W.; Xue, Y.Y. Ship's Trajectory Planning Based on Improved Multiobjective Algorithm for Collision Avoidance. *J. Adv. Transp.* **2019**, *2019*, 12. [CrossRef]
16. Zhou, K.; Chen, J.H.; Liu, X. Optimal Collision-Avoidance Manoeuvres to Minimise Bunker Consumption under the Two-Ship Crossing Situation. *J. Navig.* **2018**, *71*, 151–168. [CrossRef]
17. Liu, H.D.; Deng, R.; Zhang, L.Y. The Application research for Ship Collision Avoidance with Hybrid Optimization Algorithm. In Proceedings of the IEEE International Conference on Information and Automation (ICIA), Ningbo, China, 1–3 August 2016; pp. 760–767.
18. Li, L.; Wu, D.; Huang, Y.; Yuan, Z.-M. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Appl. Ocean. Res.* **2021**, *113*, 102759. [CrossRef]
19. Fiorini, P.; Shiller, Z. Motion Planning in Dynamic Environments Using Velocity Obstacles. *Int. J. Robot. Res.* **1998**, *17*, 760–772. [CrossRef]
20. Shen, H.; Hashimoto, H.; Matsuda, A.; Taniguchi, Y.; Terada, D.; Guo, C. Automatic collision avoidance of multiple ships based on deep Q-learning. *Appl. Ocean. Res.* **2019**, *86*, 268–288. [CrossRef]
21. Zhao, L.; Roh, M.-I. COLREGs-compliant multiship collision avoidance based on deep reinforcement learning. *Ocean Eng.* **2019**, *191*, 106436. [CrossRef]
22. Sawada, R.; Sato, K.; Majima, T. Automatic ship collision avoidance using deep reinforcement learning with LSTM in continuous action spaces. *J. Mar. Sci. Technol.* **2020**, *26*, 509–524. [CrossRef]
23. Shi, J.-h.; Liu, Z.-j. Deep Learning in Unmanned Surface Vehicles Collision-Avoidance Pattern Based on AIS Big Data with Double GRU-RNN. *J. Mar. Sci. Eng.* **2020**, *8*, 682. [CrossRef]
24. Chen, C.; Chen, X.-Q.; Ma, F.; Zeng, X.-J.; Wang, J. A knowledge-free path planning approach for smart ships based on reinforcement learning. *Ocean Eng.* **2019**, *189*, 106299. [CrossRef]
25. Xu, X.; Lu, Y.; Liu, X.; Zhang, W. Intelligent collision avoidance algorithms for USVs via deep reinforcement learning under COLREGs. *Ocean Eng.* **2020**, *217*, 107704. [CrossRef]
26. Woo, J.; Kim, N. Collision avoidance for an unmanned surface vehicle using deep reinforcement learning. *Ocean Eng.* **2020**, *199*, 107001. [CrossRef]
27. Sutton, R.S.; Barto, A.G.J. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2018; pp. 1–15.
28. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
29. Fukuto, J.; Imazu, H. New collision alarm algorithm using obstacle zone by target (OZT). *IFAC Proc. Vol.* **2013**, *46*, 91–96. [CrossRef]
30. Fossen, T.I. *Guidance and Control of Ocean Vehicles*; John Wiley & Sons Inc.: Hoboken, NJ, USA, 1994.
31. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. In Proceedings of the Presented at the Twenty-seventh Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013.
32. Van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. In Proceedings of the 30th Association-for-the-Advancement-of-Artificial-Intelligence (AAAI) Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2094–2100.
33. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay. In Proceedings of the Presented at the 4th International Conference on Learning Representations, San Juan, PR, USA, 2–4 May 2016.
34. Liu, J.; Zhao, B.; Li, L. Collision Avoidance for Underactuated Ocean-Going Vessels Considering COLREGs Constraints. *IEEE Access* **2021**, *9*, 145943–145954. [CrossRef]