Journal of
Marine Science and Engineering

MDPI

Article

A Multi-Ship Collision Avoidance Algorithm Using Data-Driven Multi-Agent Deep Reinforcement Learning

Yihan Niu [1], Feixiang Zhu [1,*], Moxuan Wei [1,†], Yifan Du [1,†] and Pengyu Zhai [2]

[1] Navigation College, Dalian Maritime University, Dalian 116026, China; niuyihan@dlmu.edu.cn (Y.N.); weimoxuan@dlmu.edu.cn (M.W.); duyifan@dlmu.edu.cn (Y.D.)
[2] School of Transportation and Logistics, Dalian University of Technology, Dalian 116024, China; zhaipengyu@mail.dlut.edu.cn
* Correspondence: zfx@dlmu.edu.cn
† These authors contributed equally to this work.

Abstract: Maritime Autonomous Surface Ships (MASS) are becoming of interest to the maritime sector and are also on the agenda of the International Maritime Organization (IMO). With the boom in global maritime traffic, the number of ships is increasing rapidly. The use of intelligent technology to achieve autonomous collision avoidance is a hot issue widely discussed in the industry. In the endeavor to solve this problem, multi-ship coordinated collision avoidance has become a crucial challenge. This paper proposes a multi-ship autonomous collision avoidance decision-making algorithm by a data-driven method and adopts the Multi-agent Deep Reinforcement Learning (MADRL) framework for its design. Firstly, the overall framework of this paper and its components follow the principle of "reality as primary and simulation as supplementary", so a real data-driven AIS (Automatic Identification System) dominates the model construction. Secondly, the agent's observation state is determined by quantifying the hazardous area. Then, based on a full understanding of the International Regulations for Preventing Collisions at Sea (COLREGs) and the preliminary data collection, this paper combines the statistical results of the real water traffic data to guide and design the algorithm framework and selects the representative influencing factors to be designed in the collision avoidance decision-making algorithm's reward function. Next, we train the algorithmic model using both real data and simulation data. Meanwhile, Prioritized Experience Replay (PER) is adopted to accelerate the model's learning efficiency. Finally, 40 encounter scenarios are designed and extended to verify the algorithm performance based on the idea of the Imazu problem. The experimental results show that this algorithm can efficiently make a ship collision avoidance decision in compliance with COLREGs. Multi-agent learning through shared network policies can ensure that the agents pass beyond the safe distance in unknown environments. We can apply the trained model to the system with different numbers of agents to provide a reference for the research of autonomous collision avoidance in ships.

Keywords: MASS; multi-ship autonomous collision avoidance decision-making; data-driven; MADRL

check for updates

Citation: Niu, Y.; Zhu, F.; Wei, M.; Du, Y.; Zhai, P. A Multi-Ship Collision Avoidance Algorithm Using Data-Driven Multi-Agent Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2023**, *11*, 2101. https://doi.org/10.3390/jmse11112101

Academic Editor: Diego Villa

Received: 23 September 2023
Revised: 23 October 2023
Accepted: 30 October 2023
Published: 1 November 2023

1. Introduction

With the boom in global maritime traffic, the number of ships is increasing rapidly. This growing trend makes maritime navigation increasingly challenging and risky. In 2021, the European Maritime Safety Agency (EMSA) counted and analyzed a total of 15,481 maritime incidents during 2014–2020, of which accidents of navigational nature (collisions, contacts, and groundings/strandings) represented 43% of all occurrences related to the ship accounted [1]. This is also the category with the largest percentage of all maritime accidents counted. Therefore, industries in the maritime sector are beginning to use intelligent technologies to achieve autonomous collision avoidance and reduce the impact of human factors on ship collision avoidance incidents.

MASS is considered to have the potential to solve the above problems in the maritime industry. Several countries and authoritative organizations have issued standards on the classification of the autonomy degree of MASS in recent years. Among them, IMO categorized the autonomy degree of MASS into four levels from a crew manning perspective at the 99th meeting of the Maritime Safety Committee (MSC 99) in 2018 [2]. This reflects a common endeavor of the shipping industry. MASS is regarded as a promising area in the maritime industry. As an important part of MASS to realize autonomous navigation tasks, ship-autonomous collision avoidance decision-making has become one of the important research issues in the field of marine engineering [3].

Research groups around the world are rapidly developing technologies with impressive results. However, most methods do not consider the coordinated or uncoordinated interaction between ships in the scenario when designing algorithms and even assume that only the own ship can take action while other target ships keep speed and course. As we know, the essence of ship collision avoidance is a continuous process of interaction between ships. Especially in multi-ship collision avoidance scenarios, the dynamic navigation status and maneuvering behavior of each ship are affected by other surrounding ships. Therefore, there is a certain gap between existing simulated scenarios and real scenarios.

This paper proposes a multi-ship distributed collision avoidance algorithm with MADRL by AIS data-driven approach, taking into consideration mixed traffic scenarios and uncoordinated scenarios in real waters. Each ship is deemed as an agent. Simulation experiments validate the effectiveness and efficiency of the algorithm in the multi-ship collision avoidance problem, which can ensure the navigation safety of ships.

The organization of this paper is stated as follows. In Section 2, we provide the literature review of ship collision avoidance decision-making. Section 3 introduces the design content and design ideas of the collision avoidance algorithm. Section 4 is the training and testing of the proposed algorithm. Section 5 is the conclusion and prospect of this paper.

2. Literature Review

Ship autonomous collision avoidance has always been a hot topic of navigation safety for smart ships. At present, the mainstream autonomous collision avoidance methods are generally divided into three categories [4].

The first category of methods is based on analytical models. This category of algorithms describes the ship's movement and its surroundings with an accurate mathematical model, such as MPC [5], VO [6,7], and APF [8]. Although these algorithms are effective, they often lack the flexibility to cope with complex and dynamic environments. For example, MPC suffers from large computational volumes and imperfect models. VO suffers from low robustness and slow processing speed. APF suffers from local optimality, external interference, and discontinuous action.

The second category of methods is based on intelligent algorithms and mainly includes the A*-based global path planning algorithm [9], Fuzzy Logic algorithm [10], and Multi-objective Evolutionary algorithm (MOEA) [11]. However, the A*-based global path planning algorithm suffers from inconsistent model prediction accuracy and lack of real-time, and MOEA suffers from difficulties in setting the objective function and non-convexity phenomena.

The third category of methods Is based on Machine Learning (ML) and mainly includes Deep Learning (DL), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL). ML and Artificial Intelligence (AI) technology are currently the most applicable methods to solve this problem [12]. For example, Wang et al. proposed a deep reinforcement learning obstacle avoidance decision-making algorithm to solve the problem of intelligent collision avoidance by unmanned ships in unknown environments. Based on the Markov Decision Process (MDP), an intelligent collision avoidance model is established for unmanned ships [13]. Sun et al. proposed an autonomous USV collision avoidance framework, DRLCA (Deep Reinforcement Learning for collision avoidance), which can be applied to USV navigation [14]. Shen et al. proposed an algorithm based on deep Q-learning for automatic collision avoidance of multiple ships, particularly which incorporates ship maneuverability, human experience, and navigation rules, and designed a restricted water test method to effectively test the capabilities of intelligent ships in a limited time frame [15]. Sawade et al. proposed a collision avoidance algorithm based on proximal policy optimization (PPO), which improves the obstacle zone by target (OZT) and enables the control of the rudder angle in continuous action space [16]. Zhao et al. proposed a DRL algorithm for ship collision avoidance based on Actor-Critic (AC), which divides the target ship area into four regions based on COLREGs and solves the case of different numbers of target ships by fixing the neural network input dimensions [17]. However, the above methods based on the single-agent concept deal with ship collision avoidance from the perspective of the own ship and do not describe the interaction behavior relations among ships directly, which is inconsistent with reality. The individual behaviors will have an impact on the overall collision avoidance result, and collision avoidance measures need to be decided in coordination with each other, especially in multi-ship collision avoidance scenarios.

Therefore, experts and research scholars have gradually extended the research direction from the single-agent system to the multi-agent system (MAS). Groups of agents within the MAS share the same environment, use sensors to perceive the environment, and take actions by using actuators. MAS usually adopts a distributed structure, which allows control authority to be distributed to the individual agents [18]. It has high reliability and robustness by using MAS to solve practical problems. However, MAS has difficulty dealing with high-dimensional continuous environments because of its concurrency. On the contrary, DRL is able to deal with high-dimensional inputs and learn to control complex actions.

MADRL combines the advantages of DRL and MAS and overcomes their inherent disadvantages. Specifically speaking, DRL models often require a large number of samples for training, and the inherent concurrency of the MAS system enables agents to generate a large amount of data concurrently, which greatly increases the number of samples, accelerates the learning process, and achieves better learning effects. At the same time, the internal structure of the neural network can solve the communication problem in MAS by using a shared policy network that exhibits implicit coordination to overcome the problem of inadequate artificially defined communication methods.

MADRL is an effective method for solving the multi-ship autonomous collision avoidance problem, which is a typical sequential decision-making process. Zhao et al. proposed a DRL-based algorithm to address the multi-ship collision avoidance problem. The algorithm adopts policy network sharing, i.e., eight ships are trained simultaneously, which improves the efficiency of policy convergence and obtains higher returns [17]. Luis et al. proposed a centralized convolutional Deep Q-network. Each agent has an ultimately independent dense layer to handle scalability [19]. Chen et al. proposed a multi-ship cooperative collision avoidance method based on the MADRL algorithm. By designing different reward weights to vary the degree of cooperation among the agents, the impact of agents in different cooperation modes on their collision avoidance behavior is discussed [20]. However, the above DRL algorithms are constructed and trained by pure simulation data. As a result, even if these models perform well in simulation environments, there is no guarantee that they will be able to make equally effective and safe decisions in real waters. Compared with simulation data, models trained by real data can not only better cope with real navigational challenges but also more deeply absorb human experience and wisdom to ensure the ship's safety and reliability in various scenarios.

The shipborne navigation aid systems, which include RADAR/ARPA, AIS, and ECDIS (Electronic Chart Display and Information System), provide the source and real data of ship collision avoidance scenarios at sea. As a requirement (part of the International Convention for Safety of Life at Sea), AIS, which should be carried for all ships from 2002, shall provide information including the ship's identity, type, position, course, speed, navigational status, and other safety-related information—automatically to appropriately equipped shore stations, other ships, and aircraft. Meanwhile, the reporting interval of AIS messages is from 2 s to 6 min, depending on the message types and the ship's dynamic conditions [21].

Growing ships have been equipped with AIS devices in the past twenty years, so a huge amount of marine traffic scenarios that are useful to develop ship autonomous collision avoidance algorithms have been recorded and accumulated in shore-based systems.

Motivated by all of the above, this paper proposes A multi-ship distributed collision avoidance algorithm with MADRL by real AIS data-driven, taking into consideration mixed traffic scenarios and uncoordinated scenarios in real waters. In this paper, the overall framework and its constituent units follow the principle of "reality as primary and simulation as supplementary", which determines that real AIS data-driven model structure occupies a dominant position. Then, we combine the statistical results of the real water traffic data to guide and design the MADRL framework and select the representative influencing factors to be designed into the collision avoidance decision-making algorithm's reward function. Next, based on the idea of "reality as primary and simulation as supplementary", the proportion of practical significance is selected to use real-AIS data and simulation data for model training, respectively. Finally, the simulation tests the collision avoidance effect of this algorithm in a library of complex and difficult ship encounter scenarios based on the idea of the Imazu problem.

3. Multi-Ship Collision Avoidance Decision-Making Algorithm Design

In this section, we will describe COLREGs, ship coordinated and uncoordinated behaviors, and design the flow chart, observation state, action space, reward function, and neural network model in the proposed algorithm.

3.1. COLREGs

In the sight of one another, overtaking situations, head-on situations, and crossing situations are three situations of encounters or three positional relationships that are constituted when two ships meet during navigation. Chapter two of COLREGs defines the conditions that constitute these three situations and also the rights and obligations of the ship in them. The situations defined by COLREGS are also the environment in which the ship's autonomous collision avoidance decision system operates as the agent. The specific definitions are shown below [22,23]:

- Rule 13 (Overtaking): If a vessel is deemed to be overtaking when coming up with another vessel from a direction more than 22.5° above her beam, the situation is considered to be overtaking. Notwithstanding anything contained in the Rules of Part B, Sections I and II, any vessel overtaking any other shall keep out of the way of the vessel being overtaken.
- Rule 14 (Head-on situation): Each ship should turn to the starboard and pass on the port side of the other ship when there is a risk of collision.
- Rule 15 (Crossing situation): If the courses of two vessels cross, the situation is considered as crossing situation; When two power-driven vessels are crossing so as to involve risk of collision, the vessel which has the other on her own starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel.

As is shown in Figure 1, the yellow region indicates the head-on situation, the red region indicates the port crossing situation, the green region indicates the starboard crossing situation, and the white region indicates the overtaking situation in which the agent ship is the overtaken vessel. In addition, the own ship (OS) is pink, and the target ship (TS) is blue.

3.2. Ship Coordinated and Uncoordinated Behaviors

The following situation may occur during the process of maned-vessel collision avoidance in the real waters: one or more vessels do not take coordinated communication or take collision avoidance actions based on COLREGs, resulting in uncoordinated collision avoidance behaviors [24]. Meanwhile, there will be a mixed traffic scenario in which manned ships and autonomous ships coexist for a certain period in the future [25]. Therefore, the possible uncoordinated behavior of all ships from the global perspective is one of the factors that MASS collision avoidance algorithms need to focus on when designing.

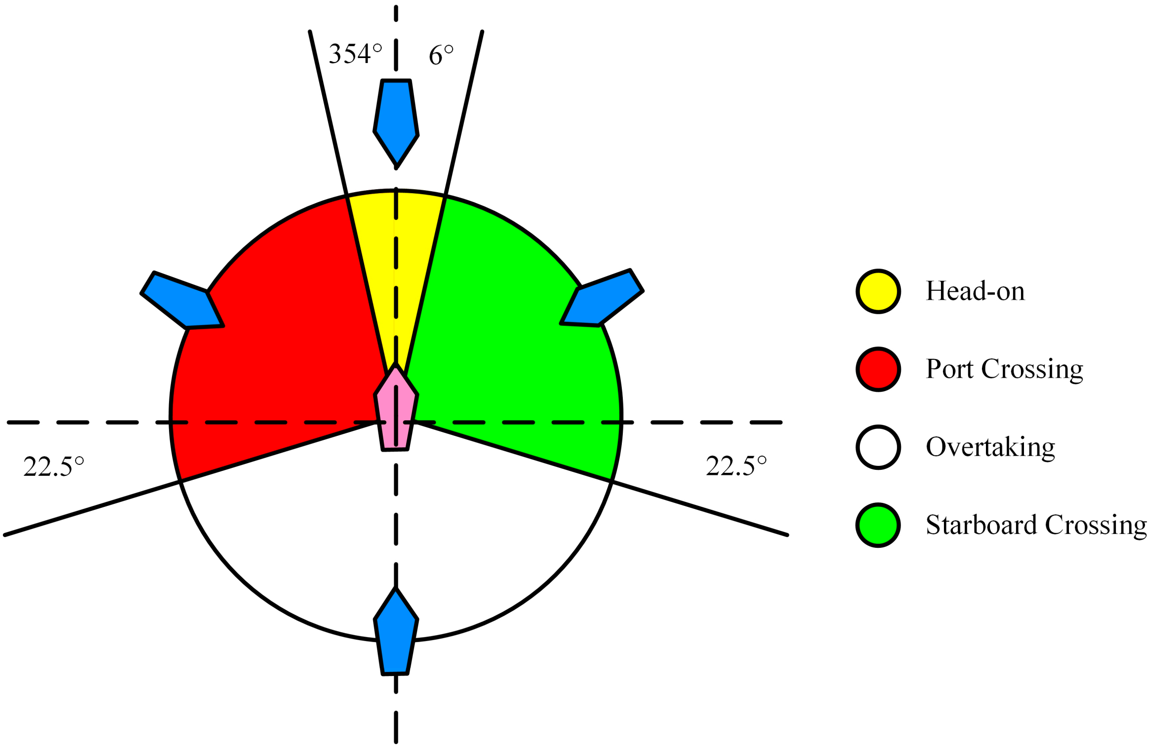


Figure 1. The typical diagram of different encounter situation.

Based on this, we define "coordinated collision avoidance behaviors" in this paper as those taken by the ship, which has the attribute of the trained agent. Specifically, the ship can take safe and rule-compliant collision avoidance decision-making measures when it recognizes a collision risk. Likewise, "uncoordinated collision avoidance behaviors" are defined as those taken by the ship which does not have the attribute of the trained agent, such as keeping speed and course without taking collision avoidance actions or taking non-rule-compliant actions.

We adopt the MAS framework, i.e., all ships within the scenario are default set as positive and rational agents that adopt coordinated collision avoidance behaviors. In order to simulate the uncoordinated scenarios in real waters, as well as to consider the sampling flexibility and enhance the model robustness factors, this paper selects the Weighted Random Sampling (WRS) method. The interval [0, 1] is divided into equal parts at interval intervals of 0.2 by the WRS method. Each interval is assigned a weight value, as shown in Table 1. A higher weight value means a higher probability that the interval will be selected.

Table 1. Selection probability of the random number generation based on WRS.

The Interval for Random Number Generation	Probability of Selecting the Interval
[0.0, 0.2]	0.10
[0.2, 0.4]	0.20
[0.4, 0.6]	0.40
[0.6, 0.8]	0.20
[0.8, 1.0]	0.10

Based on the above method, we set that there are n ships within the encounter scenario. When the $i-th$ ship decides whether to perform the coordinated collision avoidance action or not, a random number $R_i(i = 1, 2, \dots n)$ that falls within the [0, 1] probability interval will be generated. R_i represents the probability of whether the $i - th$ ship performs a collision-avoidance action or not, which can also be interpreted as the probability that the ship is given the attributes of a positive and rational agent.

In order to effectively manage the non-coordination behaviors and improve the system's overall performance, this paper proposes a flexibly adjustable non-coordination avoidance factor θ. When $R_i > \theta$, the $i - th$ ship is regarded as having the attribute of the trained agent in the collision avoidance scenario and follows the reward function design concept to positively take avoidance measures in Section 3.8. On the other hand, when $R_i < \theta$, the $i - th$ ship will no longer have the attribute of the trained agent. Specifically, the ship may keep speed and course without taking collision avoidance actions or taking non-rule-compliant actions. We set the ship's hazard recognition switch and the agent attribute switch to be mutually exclusive. When the ship recognizes a hazard, the algorithmic model will extract the failure experience or worse experience from the training experience pool. And the action space corresponding to the selected experience will be used as the action measure. This may create a more dangerous situation within the whole scenario. At this time, ships with uncoordinated behaviors will follow the new reward function, as detailed in Section 3.8.

At the same time, we can control the proportion of uncoordinated scenarios appearing by adjusting the weights of the WRS intervals and the size of the non-coordination avoidance factor θ to make the generated test scenarios as close as possible to real water. This increases the diversity and authenticity of the training data set.

3.3. Flow Chart

Figure 2 shows the flow chart of the algorithm. At the start of each cycle, state parameters are obtained, and the values of DCPA (distance of the closest point of approach), TCPA (time to the closest point of approach), distance, and bearing are calculated to obtain the current status information. Then, the risks of encounter situations are calculated during each state transfer. If there are no risks and the ship has passed and cleared the target ship, the ship will return to the planned route. If there are no risks and the ship has not passed and cleared the target ship, the ship will keep the original course and speed. If there are risks, the observation state will be calculated and input to the DDQN (Double Deep Q-Network) to make the decision. The corresponding action information is then transferred to the ship motion control system, which updates the current status information in conjunction with the ship motion model. The cycle ends when the ship reaches the planned route point or when a collision occurs with the target ship. Otherwise, the cycle will continue.

3.4. Definition of Ship Collision Avoidance Problem Based on MDP

Markov chain is a random process with Markov property, i.e., the future state depends only on the current state and is unrelated to the past state. In the ship collision avoidance problem, we can use factors such as the ship's position and speed as input states. The actions of the ship in each state are affected by certain probabilities, which can be expressed as state transfer probabilities. These describe the probability of transferring to another state in a given state.

However, the actions of ships are not only affected by the states but also by the other ships' actions in the environment, as well as the ship's desired goals. Therefore, we need to introduce the Markov Reward Process (MRP) to consider these factors. MRP is an extension of the Markov chain. It combines the probability of each state transfer with an immediate reward to take into account the effect of the particular behavior in a given state. In the ship collision avoidance problem, we can define the reward function. For example, the smaller the deviation distance, the larger the reward that the agent receives to encourage the ship to choose the appropriate actions to avoid the collision.

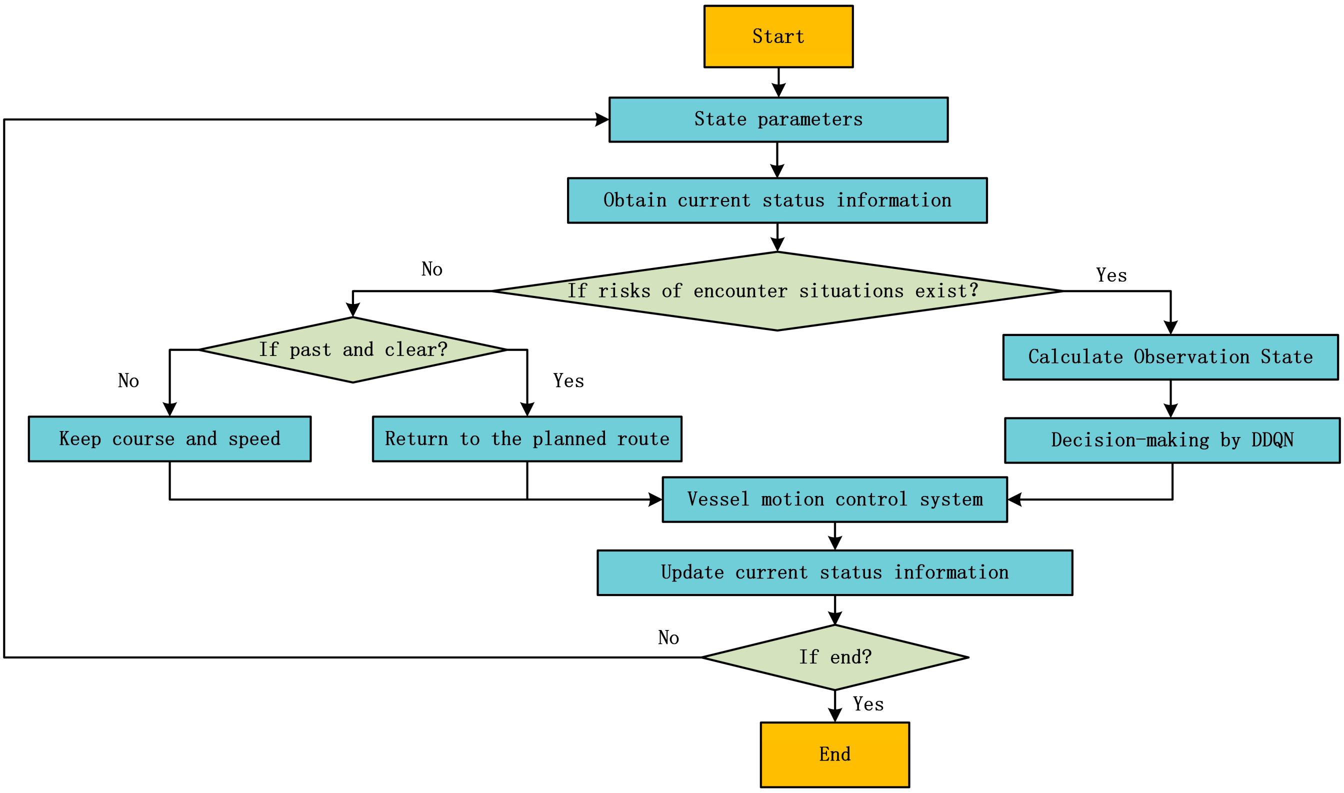

Start
State parameters
Obtain current status information
If risks of encounter situations exist?
No
Yes
If past and clear?
No
Yes
Calculate Observation State
Keep course and speed
Return to the planned route
Decision-making by DDQN
Vessel motion control system
Update current status information
No
If end?
Yes
End

Figure 2. The flow chart of autonomous collision avoidance decision-making algorithm.

On this basis, we continue to introduce decision variables that allow the ship to choose the actions under each state, thereby forming a complete MDP. At the same time, by considering all possible actions that can be taken in each state, we can establish decision rules or policies to guide the ships' actions so that the overall reward is maximized or a specific objective function is optimal.

Therefore, when applying the MADRL framework to solve the multi-ship collision avoidance decision-making problem, we describe this problem as an MDP. This method can help us to solve the ship collision avoidance problem systematically and provide guidance for decision-making. In the MDP, the agent obtains the observation state from the current environment and decides to perform the action based on it. The chosen action, in turn, indirectly affects the update of the environment and the size of the reward value. Based on the above, this paper represents the MDP as an 8-tuple $(S, O, A, \pi, P, R, \gamma, \alpha)$ as follows:

1. S is a finite set of environment states; s is the current environment state, which mainly includes ships, dynamic obstacles, static obstacles, etc.
2. O is the set of observed states of the agents; o_t is the observation state obtained by the agent in the environment at the moment t.
3. A is the action space set of the agents; a_t is the action performed by the agent at the moment t, generated by the policy function $\pi(a \mid o) = \mathbb{P}(A = a|O = o)$.
4. P is the state transfer function and $P \in [0,1]; P(s' \mid s, a) = \mathbb{P}\left(S'_{t+1} = s' \middle| S_t = s,\ A_t = a\right)$ is the probability that the state is transferred from s to $s\prime$ after the agent performs the action a_t at the moment t.
5. R is the reward function; r_t is the reward that the agent receives from the environment at the moment t.
6. γ is the decay value for future reward; α is the learning rate of the agent.

3.5. PER-DDQN

In 2015, V. Mnih's team proposed the concept of target neural networks, which officially marked the birth of DQN (Deep Q-network) [26]. Compared with traditional

Q-Learning, DQN no longer records the Q-value but uses a neural network $Q(s,a;w)$ to approximate the optimal action-value function $Q^*(s_t,a_t)$. The DQN algorithm's main advantage is its ability to deal with high-dimensional state spaces. Meanwhile, the algorithm's generalization ability can be improved through deep neural network learning to ensure scalability and applicability.

However, DQN does not guarantee that the network will always converge because DQN suffers from the maximum operator and bootstrap problems. To solve this problem, DDQN (Double Deep Q-Network) was proposed by the DeepMind team in 2016 [27]. DDQN works by setting up two independent Q-networks. One is the main neural network for selecting the maximum value action, and the other is the target neural network for evaluating this action's Q-value. The target neural network is usually a duplicate of the main network, but its parameter θ^- is not updated with each training iteration. Instead, it is copied from the main network at a certain frequency. Specifically, when we use the target network to compute the target's Q-value, the parameter θ^- is only updated once every certain number of steps so as to maintain the stability of the objective function. This results in less variation in the target value during the training process and allows for more efficient training of the primary network. At the same time, it reduces the noise and volatility in the learning process and improves the stability of training and convergence speed.

We compare the neural network performance of Nature-DQN, Target Network, and DDQN by the process of computing TD-target, as shown in Table 2.

Table 2. Comparison of three neural network constructions.

Type	Action Selection	Value Evaluation
Nature-DQN	DQN: $a^* = argmaxQ(s_{t+1},a;\theta)$	Target Network: $y_t = r_t + \gamma Q(s_{t+1},a^*;\theta)$
Target Network	Target Network: $a^* = argmaxQ\left(s_{t+1},a;\theta^-\right)$	Target Network: $y_t = r_t + \gamma Q\left(s_{t+1},a^*;\theta^-\right)$
DDQN	DQN: $a^* = argmaxQ(s_{t+1},a;\theta)$	Target Network: $y_t = r_t + \gamma Q\left(s_{t+1},a^*;\theta^-\right)$

DDQN not only alleviates the high-estimate problem but also improves usability and makes training more stable and efficient. In addition, Schaul's team proposed the Prioritized Experience Replay (PER) method in 2016 [28]. It is an enhanced experience replay method for learning by agents for training deep neural networks. It introduces the priority concept based on the traditional experience replay, i.e., it prioritizes the more important experiences for learning and makes more efficient use of the samples in the experience pool to improve the training efficiency and performance.

Based on the above, this paper adopts the PER-DDQN algorithm. It extracts all the transfer information in the experience pool that can be used for experience replication and then selects and gives priority to the transfers with a larger TD error. These experiences are more worthy of agent learning, so they are given greater priority. The model of PER-DDQN is shown in Figure 3.

Overall, the combination of DDQN and PER amplifies its intelligence advantage on a macro level, which can be understood as the agent paying more attention to failed experiences and choosing the learning order according to the experience priority. This can greatly reduce the trial-and-error process, make the network converge more quickly, and use the samples in the experience pool more efficiently to avoid experience waste. At the same time, using PER can eliminate the correlation between transitions and improve the performance of the DRL algorithm, making it more efficient and stable in dealing with complex tasks.

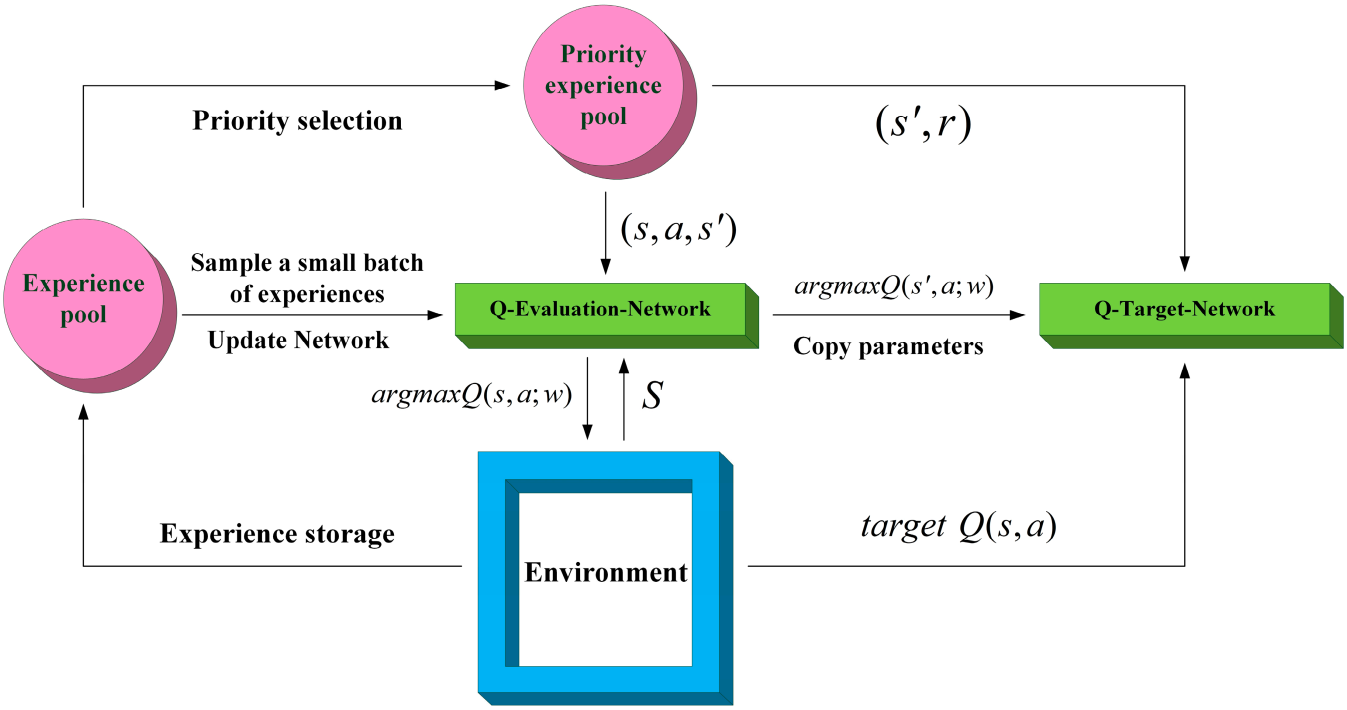


Figure 3. The model of PER-DDQN.

3.6. Observation State

In this paper, we define the distribution of MAS to constitute the set of environments as follows:

$$S = \begin{bmatrix} \psi_1 & v_1 & x_1 & y_1 \\ \psi_2 & v_2 & x_2 & y_2 \\ \ldots & \ldots & \ldots & \ldots \\ \psi_{n\text{-}1} & v_{n\text{-}1} & x_{n\text{-}1} & y_{n\text{-}1} \\ \psi_n & v_n & x_n & y_n \end{bmatrix}$$

where ψ_n is the ship's course or the dynamic obstacle's moving direction; v_n is the ship's speed or the dynamic obstacle's moving speed; x_n and y_n are the latitude and longitude of the ship or obstacle, respectively; n is the number of targets in the environment.

In past studies, research scholars have proposed many methods for predicting the hazard area of ship collision. For example, the obstacle zone by target (OZT) method based on the risk evaluation circle (REC) [29], the avoidance of bow crossing detection method [30], the predicted area of danger (PAD) model, the collision probability model, fuzzy logic and rule-based reasoning, and digital simulation. Comprehensively considering factors such as the real-time nature of environmental changes and the uncertainty of ship navigation, this paper will use an improved method based on OZT to predict the collision hazard area of each ship in the MAS.

The core idea and design principle of OZT is to “enlarge obstructions” and “advance avoidance”. Specifically, ships use sensors such as LiDAR and cameras to capture information about their surroundings, including the location, size, and shape of obstacles, which is fed into the OZT algorithm. The OZT algorithm “enlarges” the obstacle at the system's decision level; namely, the size of the obstacle is virtually magnified. Therefore, the ship's perception system will consider the obstacle to be closer than its actual distance when the ship is in close proximity to the obstacle. Ships will start to change course or slow down when they are still a certain distance away from the targets and take avoidance action in advance.

Although the OZT can allow ships to achieve certain results in avoidance actions, the method has some practical application problems. Firstly, the correct execution of OZT relies heavily on the sensors' performance. If the sensor data are inaccurate (sensor malfunction, ambient noise, obstacle occlusion, etc.), the OZT may not be able to correctly "enlarge" the obstacle, resulting in reduced avoidance performance. Secondly, OZT requires real-time environmental analysis and decision-making, which may require significant computational resources. For some unmanned systems with limited hardware resources, there may be a trade-off between OZT and other navigation tasks. Thirdly, since the design principle of OZT is "avoidance in advance", there may be the possibility of over-avoidance, which reduces the operational efficiency of the ship and the unreasonable avoidance behaviors.

Considering the above possible problems, the OZT method is improved in this paper to enhance the method's ability to cope with emergencies because the CPA (closest point of approach) is the point where two ships are closest to each other when they meet at sea. As a result, the high probability of collision in real waters is near the CPA [31]. In addition, DCPA and TCPA are CPA-derived physical quantities. DCPA is the distance between the closest approach of two ships. TCPA is the time required for a ship to reach the CPA. These parameters are very important concepts in ship collision avoidance and core indexes for developing navigation policies and assessing ship safety [32]. Therefore, the target ship's CPA is taken as the center of the circle, and the speed navigation distance (SND) R_{SND} is taken as the radius (The diameter $D_{SND} = 2R_{SND}$) to create a circular area C_1. When the ship sails to the moment t, based on the speed v of the target ship, the system calculates the distance $D_{calculation}$ that the target ship will travel in the next k set time steps (kh), and the calculation equation is shown in Equation (1).

$$D_{calculation} = khv \tag{1}$$

We extend C_1 along the direction of the target ship's course at the moment t by a distance $D_{calculation}$ to form a new circular area C_2, which is the target ship's CPA area after k time steps. As shown in Figure 4, the capsule-shaped area formed by geometrically connecting C_1 and C_2 is the collision hazard prediction area C_{OZT} set up in this paper. The length of this geometric area is $D_{Length} = D_{calculation} + D_{SND}$ and the width is $D_{width} = D_{SND}$, and all ships in the MAS should avoid entering this area. At the same time, according to the speeds of different target ships, they will be given different prediction time steps. The purpose is to control the extension distance $D_{calculation}$ unchanged so that all target ships form a collision hazard area of equal size. In this paper, we set $D_{calculation} = 1.5$ NM, $R_{SND} = 0.5$ NM, $D_{Length} = 2.5$ NM. By this way, it can balance the differences of the target ship with different features such as course, speed and size, which can reduce the algorithm's computation and facilitate the scene clustering. At the same time, the method can deal with emergencies when the sensors are faulty and prevent the observation space from generating chaos.

Considering that the input to a neural network can only be a tensor of fixed dimension, this paper designs the observation state space as an observable discretized environment and quantifies the predicted hazard area by using the grid method. This ensures that the dimension of the observation state does not change with the number of target ships in the environment. In order to be closer to the real navigational environment at sea, this grid environment uses its own perspective as the center and establishes a field of view (FOV) to detect the environment's state. At the same time, taking itself as the center of the circle, it extends outwards with a fixed value of distance interval and angle interval to form a certain number of concentric circles. In addition, we set the due north direction as the course 0°, the clockwise as the positive direction, and the angle range as 360°. The whole circumference is evenly divided by a 15° interval with a detection radius distance of 8 NM and a distance interval of 0.5 NM, as shown in Figure 5.

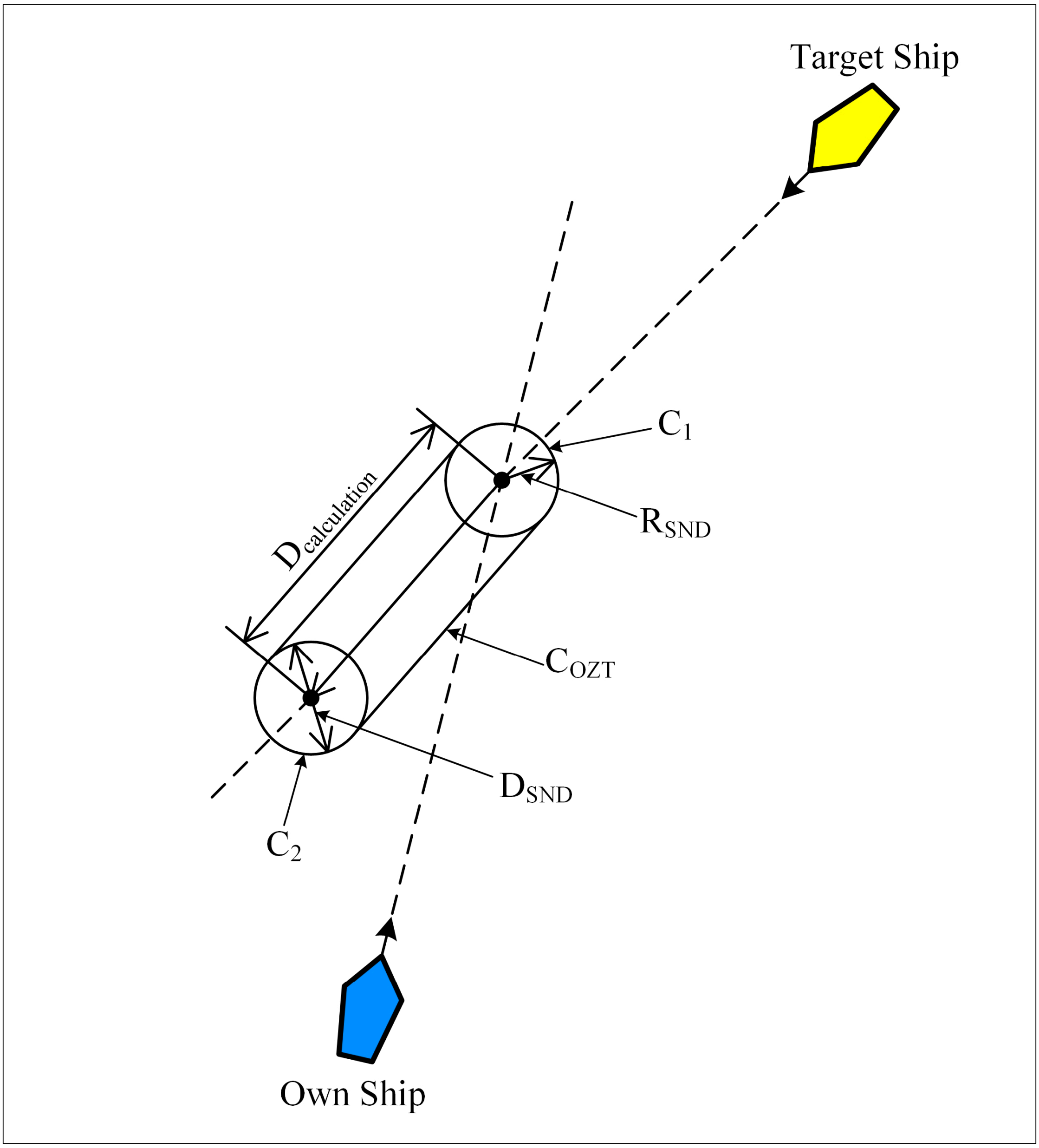


Figure 4. Improved OZT collision hazard area.

In addition, this paper defines the observation state by Boolean Operators: When the predicted collision hazard area of a ship is not in the FOV range, the ship's observation state o_t is 0; When the ship's predicted collision hazard area crosses the FOV range, the ship's observation state o_t changes to one and the collision avoidance decision-making switch is turned on. During the process of taking collision avoidance actions, the observation state o_t remains at one. The collision avoidance decision-making switch is turned off after the ship has passed and cleared the target ship. And the ship's observation state o_t becomes 0, which means the current collision avoidance task is completed.

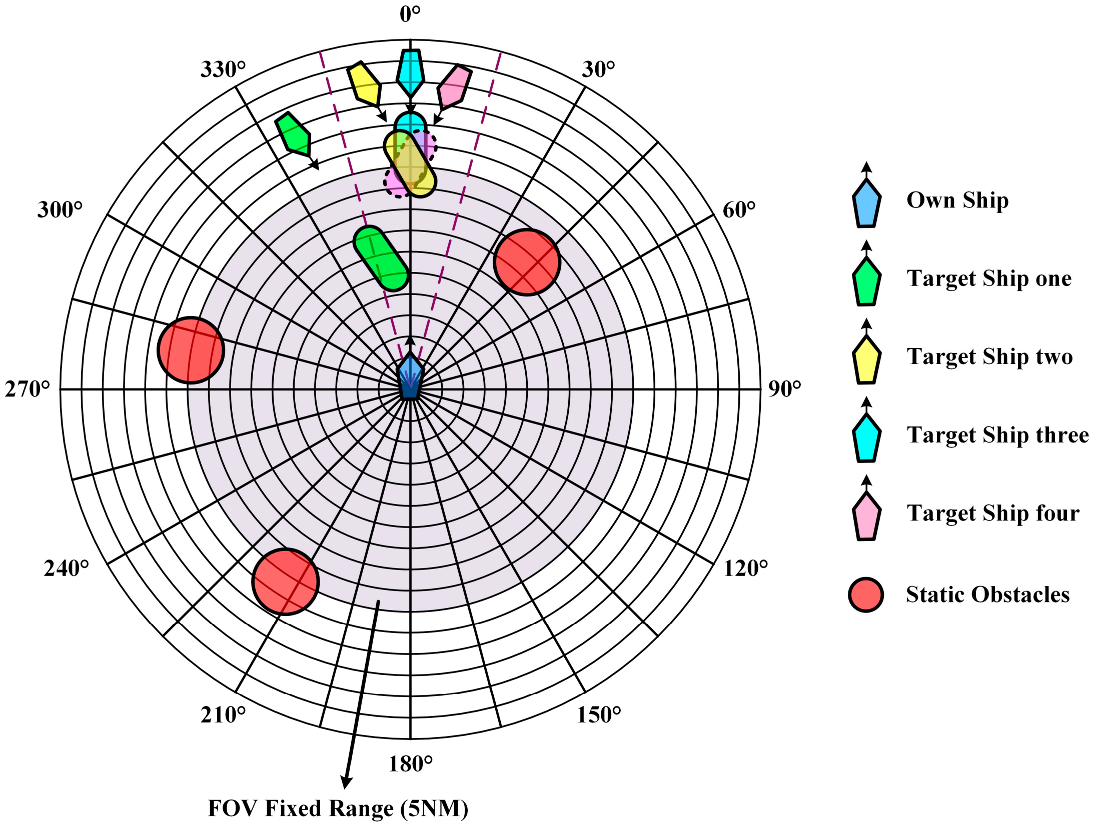


Figure 5. Observation state from the agent's own perspective.

Meanwhile, in order to reduce the input dimension of the neural network and reduce the risk of overfitting, we fixed the FOV's range and set the observation range of the agent to the environment to within 5 NM, which is helpful for us to better evaluate the generalization ability of the model. We believe that considering the partially observable perspective is an important step in the application of intelligent ships to real marine environments. At the same time, it is an effective means of replacing the state of the marine environment with areas that predict the possible risk of future collisions when we are dealing with a class of similar scenarios. In this way, similar encounter situations can be clustered and can lead to more stable decisions made by the model. By adopting the above method, the computation amount of the algorithm can be greatly reduced, and the size of the observation state space can be effectively reduced. It also prevents the observation space from generating chaotic superposition or wrong recognition of the external environment.

3.7. Action Space

Ship collision avoidance usually consists of four parts: environmental perception, taking collision avoidance action, keeping on course and speed, and returning to the planned route. In the entire collision avoidance process, the time spent on the collision avoidance decision-making (taking collision avoidance action and returning to the planned route) is much less than that spent on keeping course and speed, but it is the core part of the whole action. If the RL algorithm is used in the whole process, it will greatly increase the number of state transfers in the decision-making process, causing difficulties in model convergence. Therefore, the algorithm in this paper will only be used in the collision avoidance decision part, meaning that the agent interacts with the environment only in the collision avoidance decision-making phase, effectively shortening the number of state transitions in the MDP and substantially improving the efficiency of the algorithm. According to the above and Rule 8 [22,23]: If there is sufficient sea room, alteration alone, of course, may be the most effective action to avoid a close-quarters situation provided that it is made in good time, is substantial, and does not result in another close-quarters situation.

In collision avoidance, the pilot usually takes steering avoidance measures, including controlling the rudder angle and the course of a ship. The rudder angle change is different for different ships in the same encounter scenario. It is worth noting that the ship's course is the same at this point. Therefore, this paper will adopt the second avoidance measure as the action space, through a series of discrete course angle commands to continuously adjust the course and finally complete the ship collision avoidance. In other words, the discrete course change angle range is set as this algorithm's action space [20].

The six-degrees-of-freedom (6-DOF) model is widely used in the field of ship motion, but we usually adopt the three-degrees-of-freedom (3-DOF) model in ship collision avoidance. The 3-DOF mathematical model of a ship is shown in Figure 6.

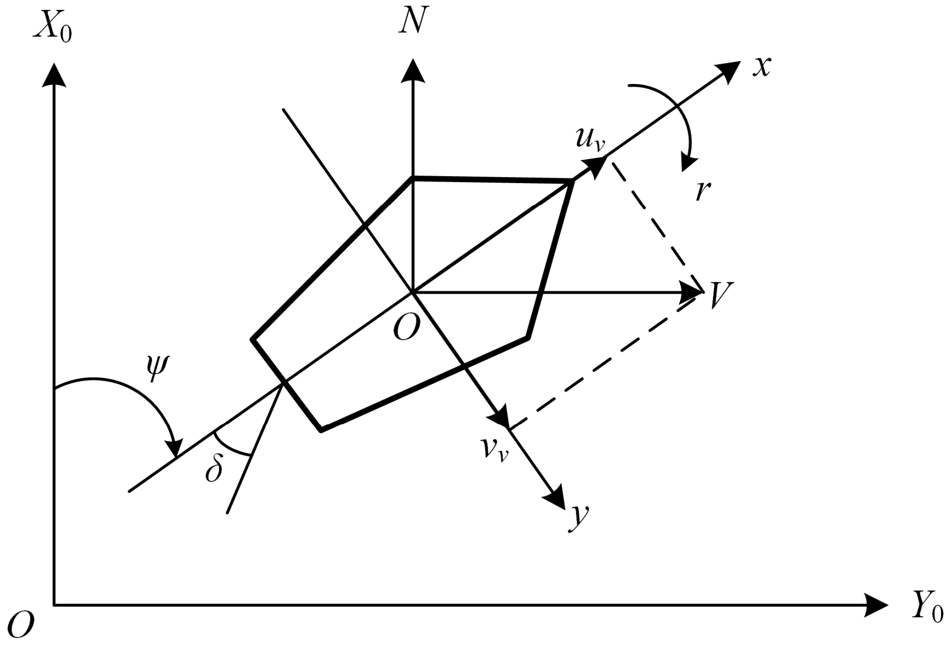


Figure 6. The 3-DOF mathematical model of a ship.

In this paper, the ship motion parameters are calculated by using Nomoto Equation [33], as expressed in Equation (2).

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} r \\ (K\delta - r)/T \\ (\delta_E - \delta)/T_E \end{bmatrix} \tag{2}$$

At the same time, the rudder angle is calculated by the PD controller and solved by the differential equation, as expressed in Equations (3)–(5).

$$r(t) = K\delta\left(1 - e^{-\frac{t}{T}}\right) \tag{3}$$

$$\psi(t) = K\delta\left(t - T + e^{-\frac{t}{T}}\right) \tag{4}$$

$$\delta(t) = K_p[\psi_c - \psi(t)] + K_d r(t) \tag{5}$$

The formula for the agent position at any moment $t_2 = t_1 + \Delta t$ is as follows:

$$x(t_2) = x(t_1) + \int_{t_1}^{t_2} v \cdot sin\psi(t_2)\, d(t) \tag{6}$$

$$y(t_2) = y(t_1) + \int_{t_1}^{t_2} v \cdot cos\psi(t_2)\, d(t) \tag{7}$$

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

This algorithm discretizes the action space and executes a series of discrete course change angle a_t commands to complete ship collision avoidance based on the collision degree hazard identification results. This paper defines that the agent turns to the left as a negative angle and the right as a positive angle. The range of discrete course change angle is $[-10°, +10°]$. The calculation formula of a ship's new course is expressed in Equation (8), and the discrete interval a_t is expressed in Equation (9).

$$\psi = \psi_{last} + a_t \tag{8}$$

$$a_t \in [-10°, -8°, -6°, -4°, -2°, 0°, +2°, +4°, +6°, +8°, +10°] \tag{9}$$

3.8. Reward Function

The agent in the RL algorithm learns by acquiring rewards through interaction with the environment and decides the appropriate action by the amount of reward value. Therefore, the reward function becomes the key to how well the agent learns. It is also the core part of the RL algorithm, which directly affects the effectiveness of the collision avoidance decision.

In order to construct a meaningful and effective reward function, this paper invests a lot of time, resources, and effort in the preliminary data collection. At the same time, considering the uncertainty of the marine environment and the diversity of navigation situations, this paper collects a large amount of relevant historical data under various types of ship navigation situations, including sailing trajectories, radar information, sensor data, and so on. By processing and integrating the collected real data, this paper analyses and clusters the data of real ship collision avoidance scenarios to reveal the correlations and trends.

Therefore, in the process of designing the reward function in this paper, the statistical results of real water traffic data are fully integrated. This is an important theoretical basis to guide the construction of the reward function so that the decisions made by the agent are closer to the results of navigation in real waters.

Combined with the COLREGs of Rule 8, Rule 16, good seamanship, expert advice, practical experience and other factors [22,23], the reward function has six main parts, as follows:

- Failure Reward: When the distance between ships is less than 0.5 NM, the algorithm defines it as a collision occurs, i.e., collision avoidance fails. Then, it will receive a larger negative reward from the environment.
- Warning Reward: When the ship moves into the collision hazard area, it will receive a small negative reward from the environment.
- Out-of-bounds Reward: When the ship enters the unplanned sea area because of taking collision avoidance actions, it will receive a medium negative reward from the environment.
- Ship Size-Sensitivity Reward: The ship's size and sensitivity can affect the ship's collision avoidance strategy and decision-making. Larger ships typically require a larger turning radius and longer braking distances, so ship size can be considered for

inclusion in the reward function. For example, larger ships could be given more success rewards based on their size and sensitivity to emphasize their collision avoidance difficulty. This can guide different types of intelligent ships to make appropriate collision avoidance decisions for themselves.

- Success Reward: When the ship successfully avoids other ships, i.e., there is no risk of collision with any other ship at the next moment, it will receive a positive reward from the environment. This reward is refined into six components by considering all factors, i.e., rule compliance, the deviation distance at the end of the avoidance, the total magnitude of ship course changes during the avoidance process, the amount of the cumulative rudder angle during the avoidance process, the DCPA when clear of the other ship and the number of rudder operations.
- Other Reward: Except for the four cases mentioned above, the agent will not receive a reward from the environment, i.e., the reward is 0.

To sum up, the definition of the reward function used in this algorithm is specified in Equations (10) and (11).

$$Reward = \begin{cases} -20 & \textit{ship collision} \\ -2 & \textit{enter the collision hazard waters} \\ -5 & \textit{enter the unscheduled waters} \\ w_1 \times w_2 \times L \times B \times D & \textit{size-sensitivity impact extent} \\ [k_1\ k_2\ k_3\ k_4\ k_5\ k_6][R_{COLREGs}\ R_{deviation}\ R_{\Delta\psi}\ R_{\delta}\ R_{DCPA} R_{rudder}]^T & \textit{reach the destination} \\ 0 & \textit{other} \end{cases} \tag{10}$$

$$\begin{array}{l} R_{COLREGs} = \begin{cases} +5, & \textit{rule compliance} \\ -5, & \textit{rule noncompliance} \end{cases} \\ R_{deviation} = \frac{d_{deviation}}{2} = \frac{d_{avoid} + d_{resumption} - d_{planned}}{2} \\ R_{\Delta\psi} = \frac{|\Delta\psi|}{10} \\ R_{\delta} = \sum\limits_{i} |\delta_i| \\ R_{DCPA} = \frac{1}{(n-1)} \sum\limits_{i}^{n-1} DCPA_i \\ R_{rudder} = (7 - n_{rudder}) \end{array} \tag{11}$$

where $d_{deviation}$ is the deviation distance at the end of the avoidance; $\Delta\psi$ is the ship's course angle during the avoidance process; δ_i is the magnitude of the $i - th$ rudder angle; n is the total number of ships in the current encounter situation; $DCPA_i$ is the distance to closest point of approach when passing and clearing the $i - th$ target ship; n_{rudder} is the total number of rudder operations; w_1 is maneuver difficulty coefficient based on the ship size; w_2 is ship maneuver sensitivity coefficient; L is the ship's length between perpendiculars; B is the ship's breadth; T is the ship's draft; k_i is the weight of each successful collision avoidance reward and $\sum_1^6 k_i = 1$, where $k_1 = 0.3$, $k_2 = 0.15$, $k_3 = 0.15$, $k_4 = 0.1$, $k_5 = 0.2$, $k_6 = 0.1$.

By selecting an action based on the above reward function, the ship is given the attribute of the trained agent and takes a coordinated collision avoidance action. However, ships with uncoordinated behaviors, as elaborated in Section 3.2, will no longer fully follow this reward function. We modify the reward function in terms of safety, rule compliance, and deviation distance, as shown in Equations (12) and (13).

$$Reward = \begin{cases} -20 & ship\ collision \\ +5 & enter\ the\ collision\ hazard\ waters \\ -5 & enter\ the\ unscheduled\ waters \\ w_1 \times w_2 \times L \times B \times D & size\text{-}sensitivity\ impact\ extent \\ [k_1\ k_2\ k_3\ k_4\ k_5\ k_6][R_{COLREGs}\ R_{deviation}\ R_{\Delta\psi}\ R_{\delta}\ R_{DCPA} R_{rudder}]^T & reach\ the\ destination \\ 0 & other \end{cases} \quad (12)$$

$$\begin{array}{l} R_{COLREGs} = \begin{cases} -5, & rule\ compliance \\ +5, & rule\ noncompliance \end{cases} \\ R_{deviation} = \frac{d_{deviation}}{2} = \frac{d_{avoid}+d_{resumption}-d_{planned}}{2} \\ R_{\Delta\psi} = \frac{|\Delta\psi|}{10} \\ R_{\delta} = \sum\limits_i |\delta_i| \\ R_{DCPA} = \frac{1}{(n-1)} \sum\limits_i^{n-1} DCPA_i \\ R_{rudder} = (7 - n_{rudder}) \end{array} \quad (13)$$

where k_i is the weight of each successful collision avoidance reward and $\sum_1^6 k_i = 1$, where $k_1 = 0.35$, $k_2 = 0.25$, $k_3 = 0.1$, $k_4 = 0.1$, $k_5 = 0.15$, $k_6 = 0.05$.

4. Training and Testing of Algorithm Model

In this paper, CPU (12th Gen Intel® Core™ i5-12400, Santa Clara, CA, USA) and GPU (Intel® UHD Graphics 730) are the equipment configurations for training and testing the algorithmic model. At the same time, pycharm software (Runtime version: 17.0.4.1) with python 3.10 is used to develop the algorithmic model.

4.1. Training Set

4.1.1. Real-Data Training Set

In this paper, the real encounter situation scenario data obtained from the literature [24] are used as the real-data training set for the algorithm. The specific approach is to screen out five groups of encounter information with different ship numbers, which are used as five units in the training set to serve the model training. And the ship's longitude and latitude information are converted to coordinate parameters in the XY coordinate system of this paper by Mercator projection so as to reproduce the real encounter scene in the training set.

We define a complete training cycle to consist of a single training session of its five constituent units. This paper completes a total of 10 training cycles and records the success rate of collision avoidance for each unit under each training cycle. We treat each unit of single training in each training cycle as an epoch, with each epoch containing n_1 iterations, and each iteration containing n_2 episodes. Each epoch trains all encounter situations (episodes) in its scene and records its training data at approximately equal intervals. At the same time, the initial value of $\varepsilon - greedy$ is defined as 0.90, increasing by 0.005 for every n_3 episodes; the neural network parameter θ_t^- is updated once for every n_4 episodes. The data information for each part of the training set is shown in Table 3.

In this paper, East is set as the positive X-axis direction, and North is set as the positive Y-axis direction in NM. The course is set using a circular representation. The results of all training cycles are shown in Figure 7. The curves represent the collision avoidance success rate of each unit in each training cycle driven by real data.

Table 3. The information of real-data training set.

Number of Ships	Episodes	n_1	n_2	n_3	n_4
Two	67,849	118	575	400	1000
Three	17,940	65	276	200	500
Four	4316	26	166	100	200
Five	951	19	50	50	50
Six	248	31	8	8	25

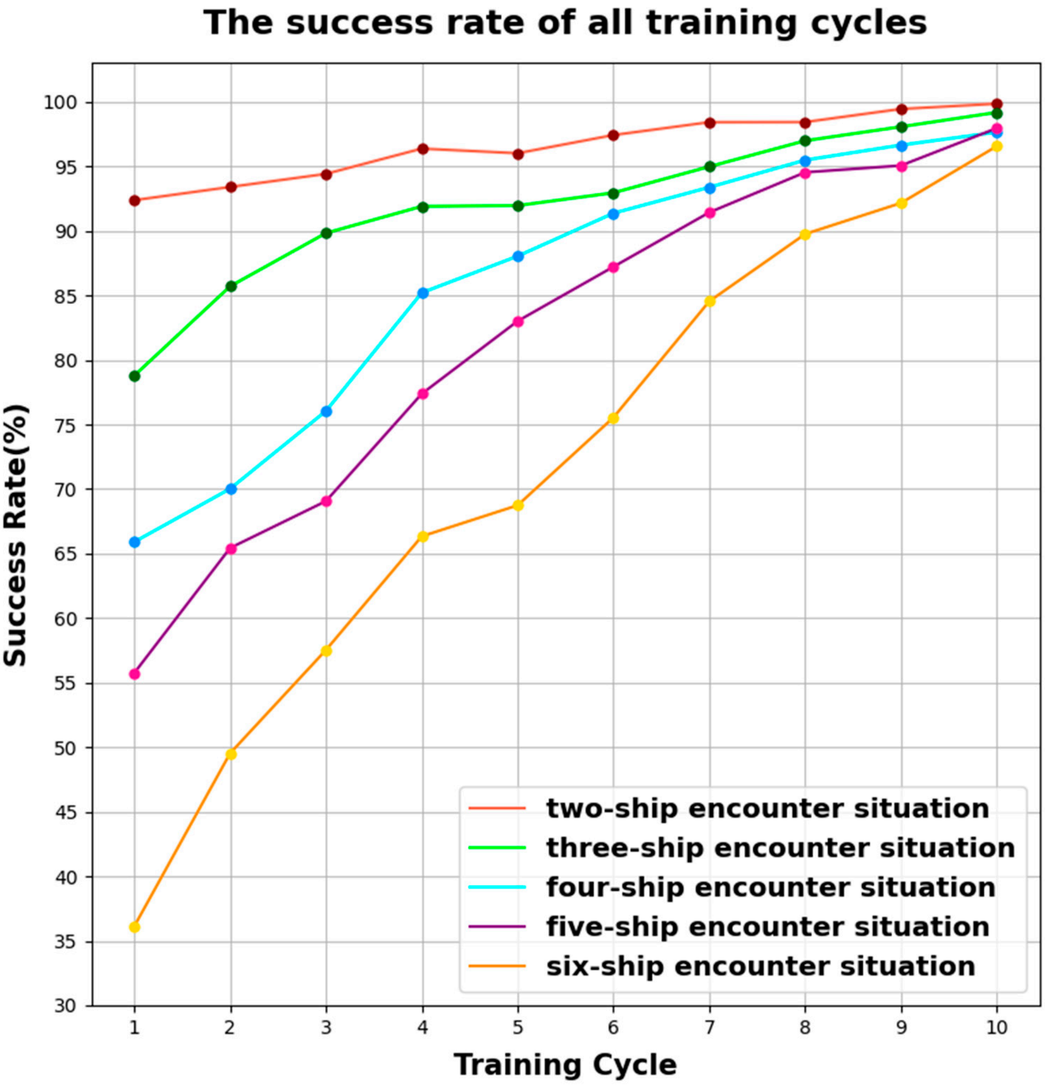


Figure 7. The real-data success rate for all training cycles.

From Figure 7, the model gradually and steadily converges in the success rate of collision avoidance with increasing training. Although there are very few cases of regression in the success rate, the success rate still shows an overall increasing trend. For situations where the number of ships is less, the success rate can increase at a steady pace with each training cycle. For situations with a high number of ships, the success rate is usually not high in the first training cycle. However, after a certain number of training cycles, the failure experience is focused on in the next learning. Therefore, the success rate of collision avoidance shows a significant increase. The greater the number of ships in the encounter situation, the faster the success rate improves.

In summary, the learning ability of the agent is gradually improved through the accumulation of training volume, and its abilities to deal with complex situations are becoming more and more strong. At the same time, the model trained by real data-driven training can ensure a high success rate when dealing with multi-ship situations. This shows that the model originated from reality and can be applied to it, which has a certain practical significance.

4.1.2. Simulation Data Training Set

The collected real-data-driven training sets do not cover all possible encounter scenarios because of high economic and time costs. Alternatively, the ship encounter scenarios are endless, and any slight change in the ship parameters will form new scenarios. And it may have an impact on the decision-making and the collision avoidance result. Although the agent's learning ability had been trained very well by real data, it may have insufficient coping ability when the agent faces unfamiliar and complex situations in the future.

Based on the above, we can conclude that it requires us to continue to enrich a large number of brand-new training scenarios so as to obtain more efficient and better training models. According to the COLREG definition of the encounter situation, we could "virtually" break the situation down into several single-ship situations under the perspective of any one ship. Therefore, we put 12 ships into the MAS. By designing the ship's course, speed, position, and destination, we make these ships constitute a variety of encounter situations, including head-on situations, port crossing situations, starboard crossing situations, overtaking situations, and overtaken situations. At the same time, considering the realism and uncertainty of the traffic flow, the weights are assigned to the integers within the interval [2,12] by the WRS method before starting the training of each episode. The larger weight value means the higher probability that the number is selected in the sample, as shown in Table 4.

Table 4. Selection probability of the ship number in the encounter scenario based on WRS.

Integer Interval Indicating the Number of Ships	Probability of Each Element in the Interval Being Selected
[2, 3]	0.05
[4, 5, 6]	0.15
[7, 8, 9]	0.10
[10, 11, 12]	0.05

Based on the above real-data training, the model can ensure a high success rate when dealing with situations with a relatively small number of ships. Therefore, situations with fewer ships will be given less weight when training on the simulation data in this subsection. This can improve learning efficiency and reduce the learning of similar experiences. At the same time, we also give less weight to encountering situations with excessive ships, such as 10, 11, and 12 ships. Although it is also achievable to successfully complete all ship collision avoidances with a certain amount of training, the real traffic flow is seldom so complex with such a large number of ships.

After selecting and determining the number of encounter situation ships i $(i = 1, 2, 3, \ldots, 12)$ in the above way, we further select the ships corresponding to the number i in the MAS with 12 ships set up by complete randomization. In this way, the initial position of the ship and the training scenario are determined. The encounter scenarios set by double random selection of the ship number and ship position can greatly enrich the diversity of the simulation data training set, which is conducive to improving the model's coping ability and learning ability.

At the same time, considering that the ship's course is not constant in the real traffic situation, it will be affected by external factors such as wind, waves, currents, etc. Therefore, this paper sets that the course of each agent will be randomly determined within $\pm 5^\circ$ of the set value. The trajectory mapping interval in the collision avoidance decision-making phase is set to 30 s, i.e., the time step of decision-making is 30 s. The information on ship navigation in the simulation-data training set is shown in Table 5. In this paper, the intelligent ship "YU KUN" is selected as the experimental model [34], and its parameters are shown in Table 6.

Table 5. Navigation information of MAS.

Ship No.	ψ (0°)	X (NM)	Y (NM)
Ship 1	[355, 5]	0.000	0.000
Ship 2	[25, 35]	−2.500	−4.330
Ship 3	[55, 65]	−6.062	−3.500
Ship 4	[85, 95]	−10.000	0.000
Ship 5	[115, 125]	−6.062	3.500
Ship 6	[145, 155]	−5.000	8.660
Ship 7	[175, 185]	0.000	8.000
Ship 8	[205, 215]	3.000	5.196
Ship 9	[235, 245]	8.660	5.000
Ship 10	[265, 275]	8.000	0.000
Ship 11	[295, 305]	6.062	−3.500
Ship 12	[325, 335]	4.500	−7.794

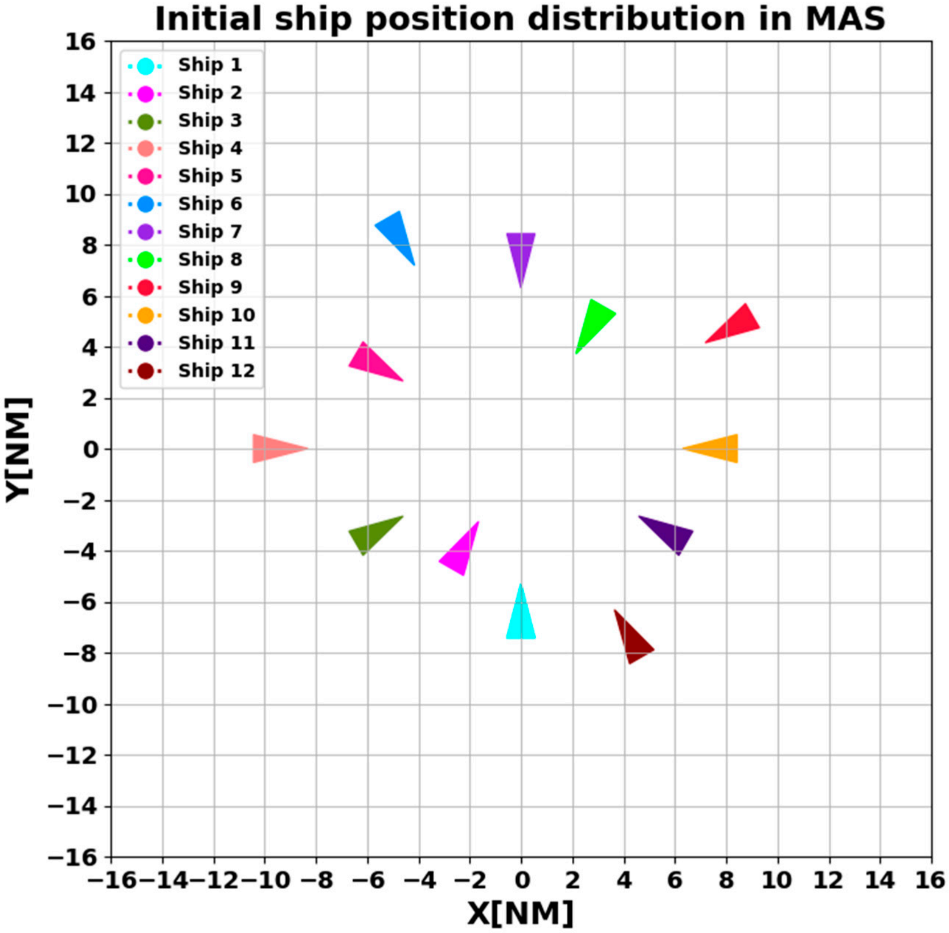


Figure 8. The initial ship position distribution in MAS.

Table 6. Ship parameters of the "YU KUN".

Physical Quantity	Symbol	Numerical Value
Length between perpendiculars (m)	L	105
Breadth (m)	B	18
Speed (kn)	V	12
Draft (m)	D	5.4
Turning ability index (1/s)	K	−0.2257
Following index (s)	T	86.8150
Controller gain coefficient (-)	K_p	2.2434
Controller differential coefficient (-)	K_d	35.9210

$LBVDKTK_pK_d$The initial ship position distribution in MAS is shown in Figure 8.

In addition, this paper follows the principle of "reality as primary and simulation as supplementary" to set up the total training set, and its content composition is shown in Figure 9.

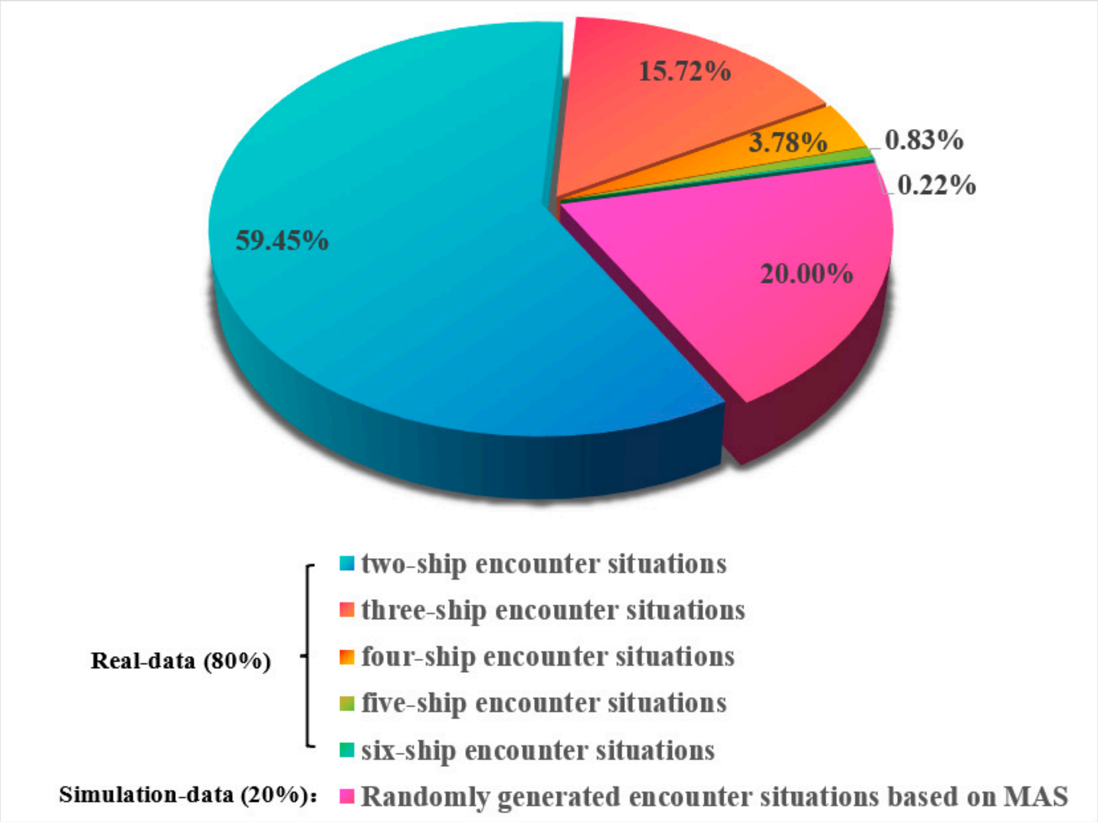


Figure 9. Content composition of the total training set.

From Figure 9, it can be seen that the real-data set in the previous subsection occupies 80% of the total training set, with a total of 913,040 episodes trained in 10 training cycles. And the remaining 20% has the simulation-data training set of this subsection constituting. In this part of the training set, each multi-ship encounter scenario has randomly generated ships in the MAS. We randomly generated 22,826 episodes by the method of WRS described above. The information and collision avoidance success rate of each episode is recorded and used as a complete training cycle (training subset). After that, this training subset was continued to repeat nine times without changing any of the training parameters, and the success rate of collision avoidance was recorded. Because each training cycle contains a sufficiently large number of episodes, and they are all generated in a random manner with a certain level of complexity. The resulting large number of random training samples is meaningful for both the improvement of the model generalization ability and the applicability extension.

Likewise, the idea of parameter setting in this subsection is consistent with the real data set. We treat each unit of single training in each training cycle as an epoch, with each epoch containing 115 iterations and each iteration containing 200 episodes. Each epoch trains all encounter situations (episodes) in its scene and records its training data at approximately equal intervals. At the same time, the initial value of $\varepsilon - greedy$ is defined as 0.90, increasing by 0.005 for every 1000 episodes; the neural network parameter θ_t^- is updated once every iteration. The results of all training cycles are shown in Figure 10. The curves represent the collision avoidance success rate of each unit in each training cycle driven by simulation data.

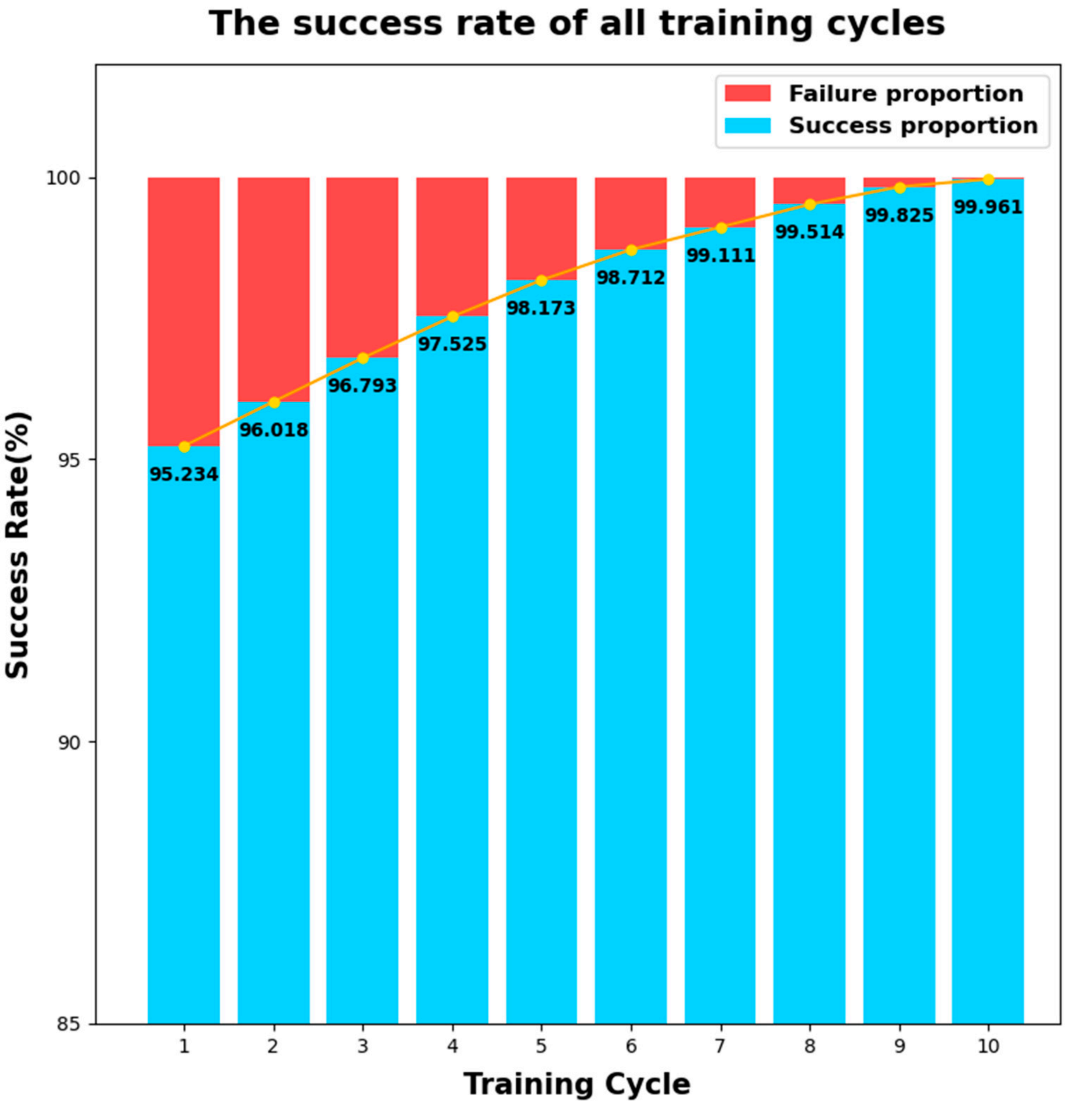


Figure 10. The simulation-data success rate for all training cycles.

In combination with the model training process and Figure 10, we can find that the model may fail the first few times in complex encounter situations. However, the model uses the PER technique and always follows the principle of "scenario adaptation" when constructing encounter scenarios. Therefore, after continuous focused learning, the agent can make the model converge quickly and stably in situations where the number of ships is "moderate", such as four ships, five ships . . . eight ships, etc. At the same time, the model performs excellently and can be trained successfully for all episodes in most iterations. It was even able to gradually optimize the navigation process based on successful collision avoidance.

4.2. Testing Set

In the autonomous ship navigation field, the Imazu problem is widely considered a series of navigational collision avoidance challenges. In order to verify the algorithm's effectiveness and the model's generalization ability, this section designs and extends 40 scenarios as the encounter scenario library based on the Imazu problem's idea. The encounter scenario library includes relatively difficult and very difficult scenarios as a way to verify the model's expressiveness and usefulness in complex environments. The idea of building this scenario library mainly stems from the following aspects:

- Comprehensiveness extension: By testing to include a variety of possible real-world sailing scenarios, we can ensure that the algorithm is able to cope with the challenges in various aspects of actual sailing;
- Improving the model's generalization ability: Diversified scenarios can help the model learn richer data, thus making its performance more stable and reliable in unknown environments;
- Simulating extreme situations: The particularly difficult scenarios in the encounter scenario library can simulate extreme situations that might be encountered in reality, which is essential for assessing the model's performance under stress;
- Enhancing verification credibility: By verifying the model's performance in various scenarios, we can more confidently ensure its safety and effectiveness in real-world applications.

Overall, the encounter scenario library has been built to provide a comprehensive, practical, and challenging test environment to ensure the wide applicability of the model. By verifying in such a scenario library, the model not only demonstrates its excellent performance in complex environments but also further ensures its usefulness and safety. The initial information of the scenario library is shown in Table 7. Where Cases 1–4 are two-ship encounter situations, Cases 5–14 are three-ship encounter situations, Cases 15–31 are four-ship encounter situations, Cases 32–36 are five-ship encounter situations, and Cases 37–40 are six-ship encounter situations. The schematic of each scenario is shown in Figure 11. The agent model is still set to the "YU KUN" with a speed of 12 kn, and the overtaken ship's speed is set to 8 kn. The non-coordination avoidance factor θ is set to 0.5. Like Section 4.1.2, the trajectory mapping interval in the collision avoidance decision-making phase is set to 100 s.

Considering the large number of figures in the test results, we structured the article by including the figures in Appendix A. The model test results are shown in Figures A1–A3. Figure A1 shows the ship's trajectory, where the initial position of each agent is represented by a different triangle, the destination is represented by the circle of the corresponding color, and the trajectory's color is the same as that of the agent in the legend. It is assumed that Agent ship one is the perspective of the own ship, and Figure A2 shows the distance change between the own ship (ship one) and the target ships under this perspective, where the dotted line 0.5 NM represents the minimum encounter distance for the urgent situation specified in this paper. Figure A3 shows the minimum passing distance of each agent ship from other agent ships.

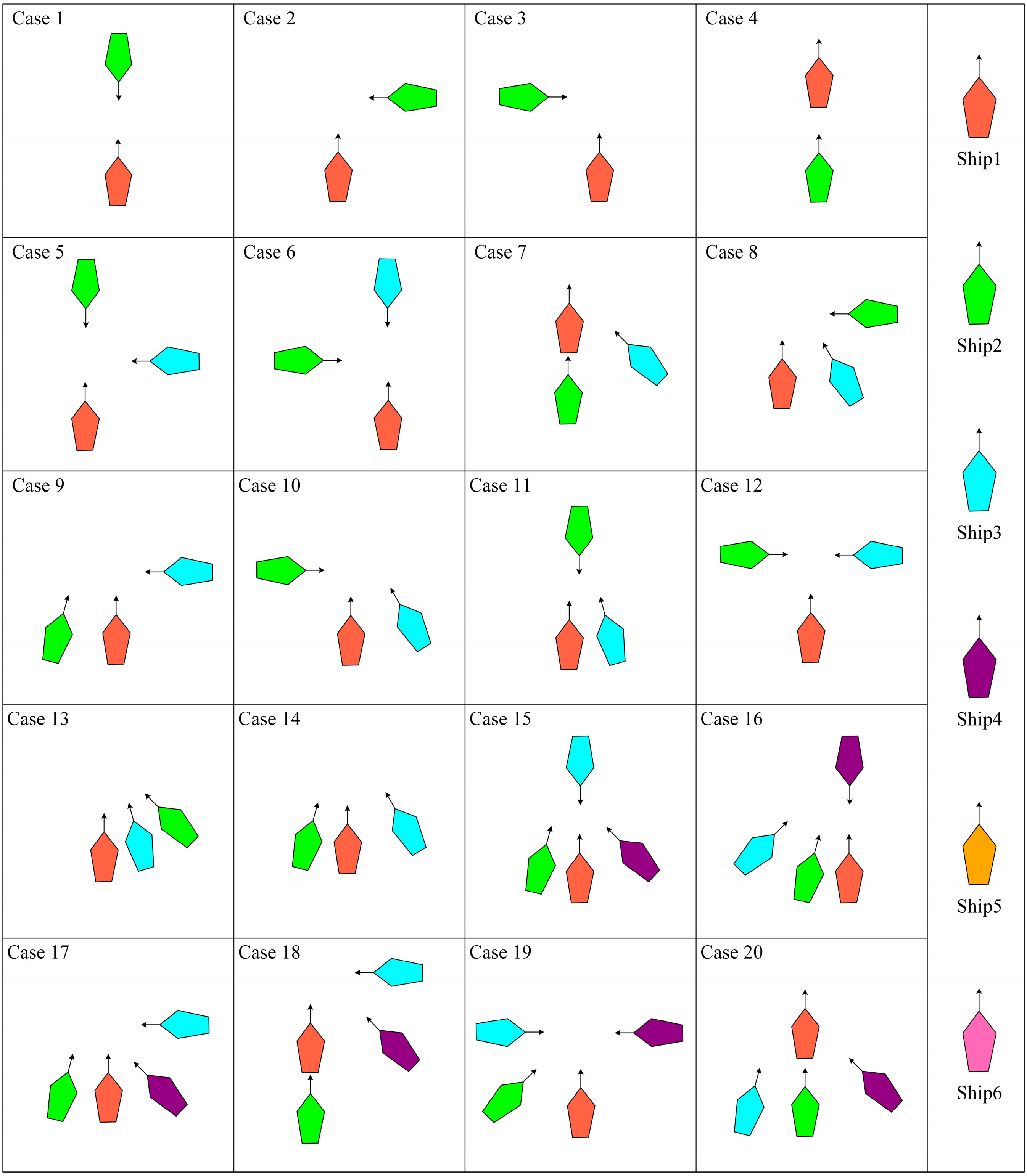
In order to better analyze the process of collision avoidance actions of each ship, Case 35 is used as an example to elaborate the whole sequential decision-making process in detail. Figure 12 shows the ships' trajectories.

Table 7. The initial information of the scenario library.

Case No.	Ship 1			Ship 2			Ship 3			Ship 4			Ship 5			Ship 6		
	X	*Y*	ψ (°)	*X*	*Y*	ψ (°)	*X*	*Y*	ψ (°)	*X*	*Y*	ψ (°)	*X*	*Y*	ψ (°)	*X*	*Y*	ψ (°)
1	0.000	−6.000	000	0.000	6.000	180	-	-	-	-	-	-	-	-	-	-	-	-
2	0.000	−6.000	000	6.000	0.000	270	-	-	-	-	-	-	-	-	-	-	-	-
3	0.000	−6.000	000	−6.000	0.000	090	-	-	-	-	-	-	-	-	-	-	-	-
4	0.000	−6.000	000	0.000	10.000	000	-	-	-	-	-	-	-	-	-	-	-	-
5	0.000	−6.000	000	0.000	6.000	180	6.000	0.000	270	-	-	-	-	-	-	-	-	-
6	0.000	−6.000	000	−6.000	0.000	090	0.000	6.000	180	-	-	-	-	-	-	-	-	-
7	0.000	−6.000	000	0.000	−10.000	000	5.657	5.657	315	-	-	-	-	-	-	-	-	-
8	0.000	−6.000	000	6.000	0.000	270	3.000	−5.196	330	-	-	-	-	-	-	-	-	-
9	0.000	−6.000	000	−1.553	−5.796	015	6.000	0.000	270	-	-	-	-	-	-	-	-	-
10	0.000	−6.000	000	−5.000	0.000	090	3.000	−5.196	330	-	-	-	-	-	-	-	-	-
11	0.000	−6.000	000	0.000	7.000	180	1.553	−5.796	345	-	-	-	-	-	-	-	-	-
12	0.000	−6.000	000	−6.000	0.000	090	6.000	0.000	270	-	-	-	-	-	-	-	-	-
13	0.000	−6.000	000	4.243	−4.243	315	1.553	−5.796	345	-	-	-	-	-	-	-	-	-
14	0.000	−6.000	000	−1.553	−5.796	015	3.000	−5.196	330	-	-	-	-	-	-	-	-	-
15	0.000	−6.000	000	−1.553	−5.796	015	0.000	6.000	180	4.243	−4.243	315	-	-	-	-	-	-
16	0.000	−6.000	000	−1.553	−5.796	015	−4.243	−4.243	045	0.000	6.000	180	-	-	-	-	-	-
17	0.000	−6.000	000	−1.553	−5.796	015	6.000	0.000	270	4.243	4.243	315	-	-	-	-	-	-
18	0.000	−6.000	000	0.000	−10.000	000	6.000	0.000	270	4.243	−4.243	315	-	-	-	-	-	-
19	0.000	−6.000	000	−4.243	−4.243	045	−6.000	0.000	090	6.000	0.000	270	-	-	-	-	-	-
20	0.000	−6.000	000	0.000	−10.000	000	−1.553	−5.796	015	4.243	−4.243	315	-	-	-	-	-	-
21	0.000	−6.000	000	4.243	4.243	225	3.000	−5.196	330	1.553	−5.796	345	-	-	-	-	-	-
22	0.000	−6.000	000	−1.553	−5.796	015	4.243	4.243	225	1.553	−5.796	345	-	-	-	-	-	-
23	0.000	−6.000	000	0.000	−10.000	000	6.000	0.000	270	3.000	−5.196	345	-	-	-	-	-	-
24	0.000	−6.000	000	−1.553	−5.796	015	6.000	0.000	270	1.553	−5.796	345	-	-	-	-	-	-
25	0.000	−6.000	000	0.000	−10.000	000	6.000	0.000	270	3.000	−5.196	330	-	-	-	-	-	-
26	0.000	−6.000	000	−1.553	−5.796	015	0.000	4.000	180	1.553	−5.796	345	-	-	-	-	-	-
27	0.000	−6.000	000	2.000	−4.000	000	0.000	6.000	180	2.000	8.000	180	-	-	-	-	-	-
28	0.000	−6.000	000	−3.000	5.196	150	6.000	0.000	270	3.000	−5.196	330	-	-	-	-	-	-
29	0.000	−6.000	000	−3.000	5.196	150	6.000	0.000	270	1.553	−5.796	345	-	-	-	-	-	-
30	0.000	−6.000	000	−1.553	−5.796	015	−3.000	−5.196	030	3.000	−5.196	330	-	-	-	-	-	-
31	0.000	−6.000	000	0.000	6.000	180	3.000	−5.196	330	1.553	−5.796	345	-	-	-	-	-	-

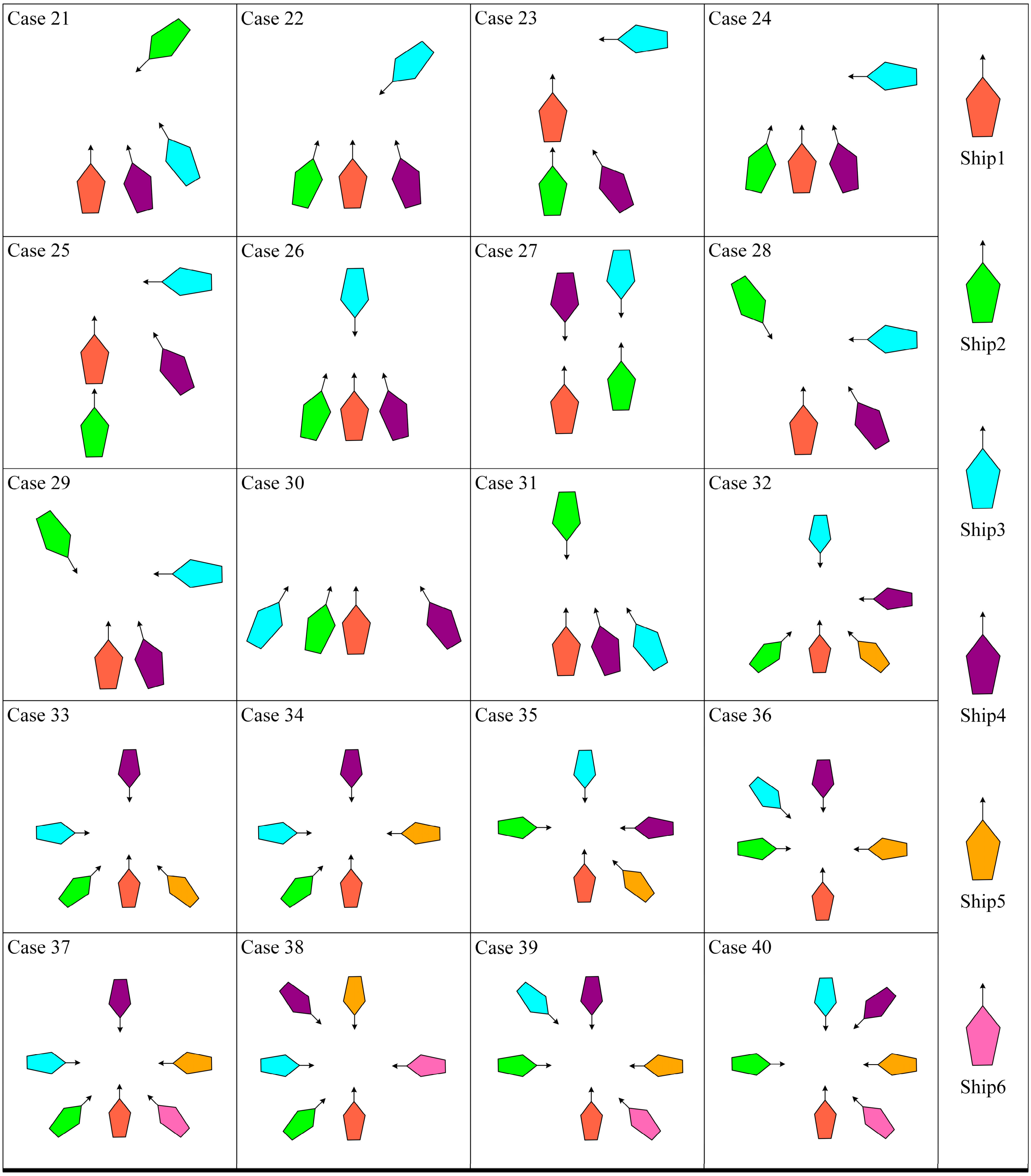
Table 7. *Cont.*

Case	**Ship 1**			**Ship 2**			**Ship 3**			**Ship 4**			**Ship 5**			**Ship 6**		
No.	***X***	***Y***	***ψ* (°)**	***X***	***Y***	***ψ* (°)**	***X***	***Y***	***ψ* (°)**	***X***	***Y***	***ψ* (°)**	***X***	***Y***	***ψ* (°)**	***X***	***Y***	***ψ* (°)**
32	0.000	−6.000	000	−4.243	−4.243	045	0.000	6.000	180	6.000	0.000	270	4.243	−4.243	315	-	-	-
33	0.000	−6.000	000	−4.243	−4.243	045	−6.000	0.000	090	0.000	6.000	180	4.243	−4.243	315	-	-	-
34	0.000	−6.000	000	−4.243	−4.243	045	−6.000	0.000	090	0.000	6.000	180	6.000	0.000	270	-	-	-
35	0.000	−6.000	000	−6.000	0.000	090	0.000	6.000	180	6.000	0.000	270	4.243	−4.243	315	-	-	-
36	0.000	−6.000	000	−6.000	0.000	090	−4.243	4.243	135	0.000	6.000	180	6.000	0.000	270	-	-	-
37	0.000	−6.000	000	4.243	−4.243	045	−6.000	0.000	090	0.000	6.000	180	6.000	0.000	270	4.243	−4.243	315
38	0.000	−6.000	000	−4.243	−4.243	045	−6.000	0.000	090	−4.243	4.243	135	0.000	6.000	180	6.000	0.000	270
39	0.000	−6.000	000	−6.000	0.000	090	−4.243	4.243	135	0.000	6.000	180	6.000	0.000	270	4.243	−4.243	315
40	0.000	−6.000	000	−6.000	0.000	090	0.000	6.000	180	4.243	4.243	225	6.000	0.000	270	4.243	−4.243	315



(**a**)

Figure 11. *Cont.*



(**b**)

Figure 11. The extended encounter scenario library based on Imazu problem. (**a**) Case 1–20 of the extended encounter scenario library; (**b**) Case 21–40 of the extended encounter scenario library.

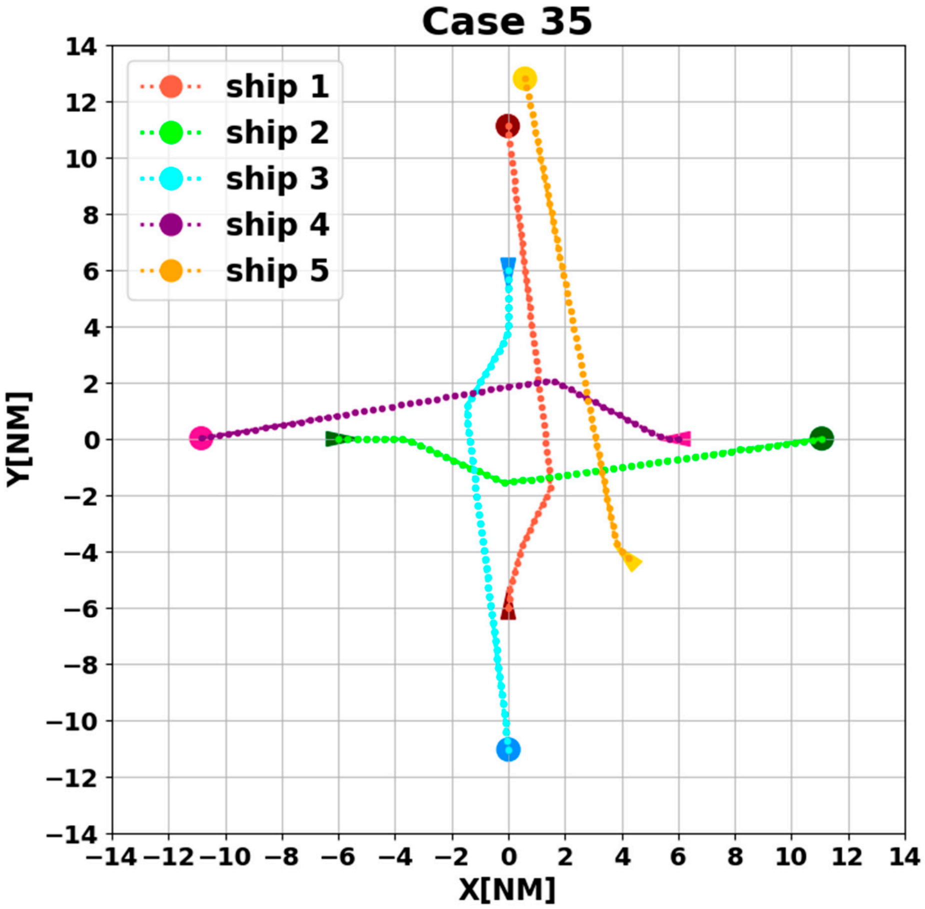


Figure 12. Ships' trajectories of Case 35.

At the initial moment, the five ships in the scenario constitute a relatively complex collision hazard situation. We split the current situation according to COLREGs and found that each ship has more than one encounter situation with other ships. For example, ship three forms the head-on situation with ship one and the crossing situation with the remaining ships, respectively.

We illustrate the working principle of the algorithmic MDP tuple by the motion process of ship three as follows. The collision avoidance algorithm model successively generates four MDP transitions (S,O,A,π,P,R,γ,α) for ship three.

1. At t = 0–600 s, ship three does not perceive a hazard in the environment, the observed state o_t is 0, and the ship is sailing towards its destination on the prescribed course;
2. At t = 600 s, ship three recognizes the hazard in the environment, at which time the observation state o_t changes to one, and collision avoidance action is started. The algorithmic model selects $a_1 = +10°$, $a_2 = +10°$, $a_3 = +10°$, $a_4 = +4°$ sequentially as actions in the action space based on the policy function π;
3. Until t = 1700 s, the ship removes the collision hazard by four course changes. At the same time, the observation state o_t becomes 0. The collision avoidance decision-making switch is turned off, and the ship starts to return to the planned route;
4. At t = 5400 s, all ships arrive at their destinations, and the sailing missions are over. The minimum passing distances of each ship from other ships are respectively 1.86 NM, 2.11 NM, 2.14 NM, 2.09 NM, and 1.86 NM. All ships are guaranteed to complete the collision avoidance decision-making beyond the safe distance.

At the same time, we can observe the agent attributes of ships in Figure 12. For example, ship five has chosen to sail around to the right instead of crossing the possible routes of the other four ships. In addition, ship two and ship four constitute the head-on situation. They are able to complete the collision avoidance task in a rule-compliant situation and do not generate extreme collision avoidance options. And the ships can resume navigation in time to avoid generating excessive deviation distances. All of the above fully reflects the core design ideas of the algorithm's reward function to focus on safety and high efficiency.

4.3. Analysis of Experimental Results

In order to clearly observe the ship's collision avoidance, we have made the colors of the ship's trajectory, distance change, and the minimum passing distance in the above figures the same as the ship's colors set in the encounter scenario library.

Among them, Figure A1 shows the navigation position and motion trajectory of each ship at different moments. We can see each ship's navigation process, including recognizing the collision risk, taking collision avoidance action, sailing to clear and past, returning to the planned route, and continuing to the destination. This is also a complete collision avoidance decision-making process. However, a ship does not completely eliminate the collision hazard through a single collision avoidance decision-making process. In general, many ships need to take several continuous steering actions in order to remove the current hazard. While some ships will still face new collision hazards during the resumption process, thus starting a new collision avoidance decision-making process. In Figure A2, this paper takes the perspective of ship one as an example. We can see that the overall trend of distance change between ships at different moments is to gradually become closer and then further away. This shows that ships can take timely collision avoidance actions after recognizing the risk so that the distances between ships are constantly moving towards higher safety. In Figure A3, we count the minimum passing distance of each agent ship from other agent ships in the complete time step. We can find that the minimum passing distance is usually presented in pairs. And this algorithm can ensure that each ship completes collision avoidance beyond the safety distance at different moments.

The results of 40 group simulation experiments show that, on the one hand, the algorithm shows sufficient coordination in unknown, diverse, and complex environments; on the other hand, it is demonstrated that the algorithm's model can be trained to full convergence through a shared policy network. Meanwhile, the trained model can be copied to the MAS with different numbers of ships to complete the collision avoidance decision-making.

5. Conclusions

This paper proposes a multi-agent collision avoidance algorithm based on DDQN and incorporates the PER technique.

Firstly, the research idea of this paper is established. The overall framework of this paper and its components follow the principle of "real data-driven as primary, simulation-driven as supplementary", so real AIS data-driven dominates the model construction. Secondly, the agent's observation state is determined by quantifying the hazardous area. Identifying the external environment from the perspective of any ship, scene clustering of target ships with similarly predicted collision hazard areas within a certain range can obtain the same observation state, effectively reducing the size of the observation state space. Then, ship-coordinated and uncoordinated behaviors are defined. In order to simulate uncoordinated scenarios in real waters, this paper proposes a non-coordination avoidance factor to decide whether to give attributes to ship intelligence or not. Thereby, the idea of multi-ship distributed collision avoidance considering the uncoordinated behaviors of the target ship is added to this paper. Next, based on a full understanding of COLREGs and the preliminary data collection, this paper combines the statistical results of the real water traffic data to guide and design the MADRL framework and selects the representative influencing factors to be designed into the collision avoidance decision-making algorithm's reward function. Subsequently, we divide the total training set of this model into two parts: one is the real data training set, and the other is the simulation data training set. Based on the idea of "reality as primary and simulation as supplementary" in this paper, the former consists of five parts of real water data, and its proportion is set to be 80% of the total training set; at the same time, this paper adopts the model of "YU KUN" for simulation and designs a MAS with 12 ships based on the ship encounter scenarios classified by COLREGs. Before each training model, the MAS will select the number of ships and their positions to complete the scenario construction by double randomization. The proportion of this part is set to be 20% of the total training set. Finally, 40 encounter scenarios are designed and extended to verify the algorithm performance based on the idea of the Imazu problem. The experimental results show that the algorithm proposed in this paper can solve the multi-ship collision avoidance problem in multiple scenarios quite efficiently. The algorithm improves the safety of autonomous ship navigation and provides a reference idea for the research of autonomous ship collision avoidance.

At present, the MADRL application in the ship collision avoidance field is still in its infancy, and the applicable conditions of this algorithm still need to be further improved. For example, the agent uses the recognition function in a way that treats other agents more as part of the environment. Such a way of coordination is obviously implicit, and the communication is not sufficient. This may lead to an unstable learning state of agents, slow convergence of the algorithms, etc. Therefore, in the next research, we will focus on achieving a more specific and efficient recognition function of agents, i.e., we will delve into the explicit method of coordinated communication among multiple agents. Meanwhile, a self-supervision mechanism can be added to the original algorithm. The aim is to better supervise the decision-making behaviors made by the agents themselves, as well as to continuously further improve the algorithm's practicality.

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

Funding: This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1601505, the National Natural Science Foundation of China under Grant 52231014, and the Liaoning Provincial Shipping Joint Fund under Grant 2020-HYLH-28.

Institutional Review Board Statement: Not applicable.

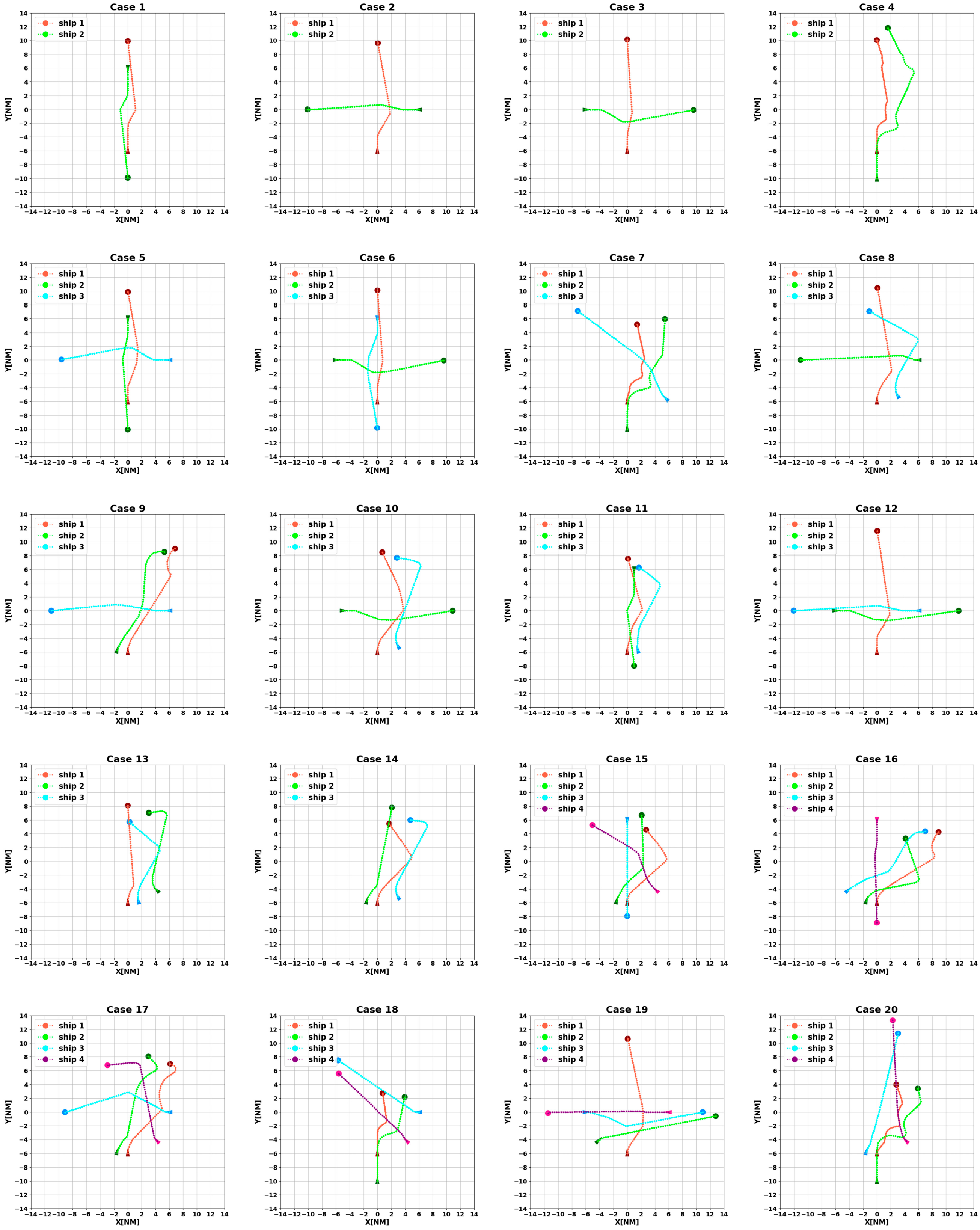
Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

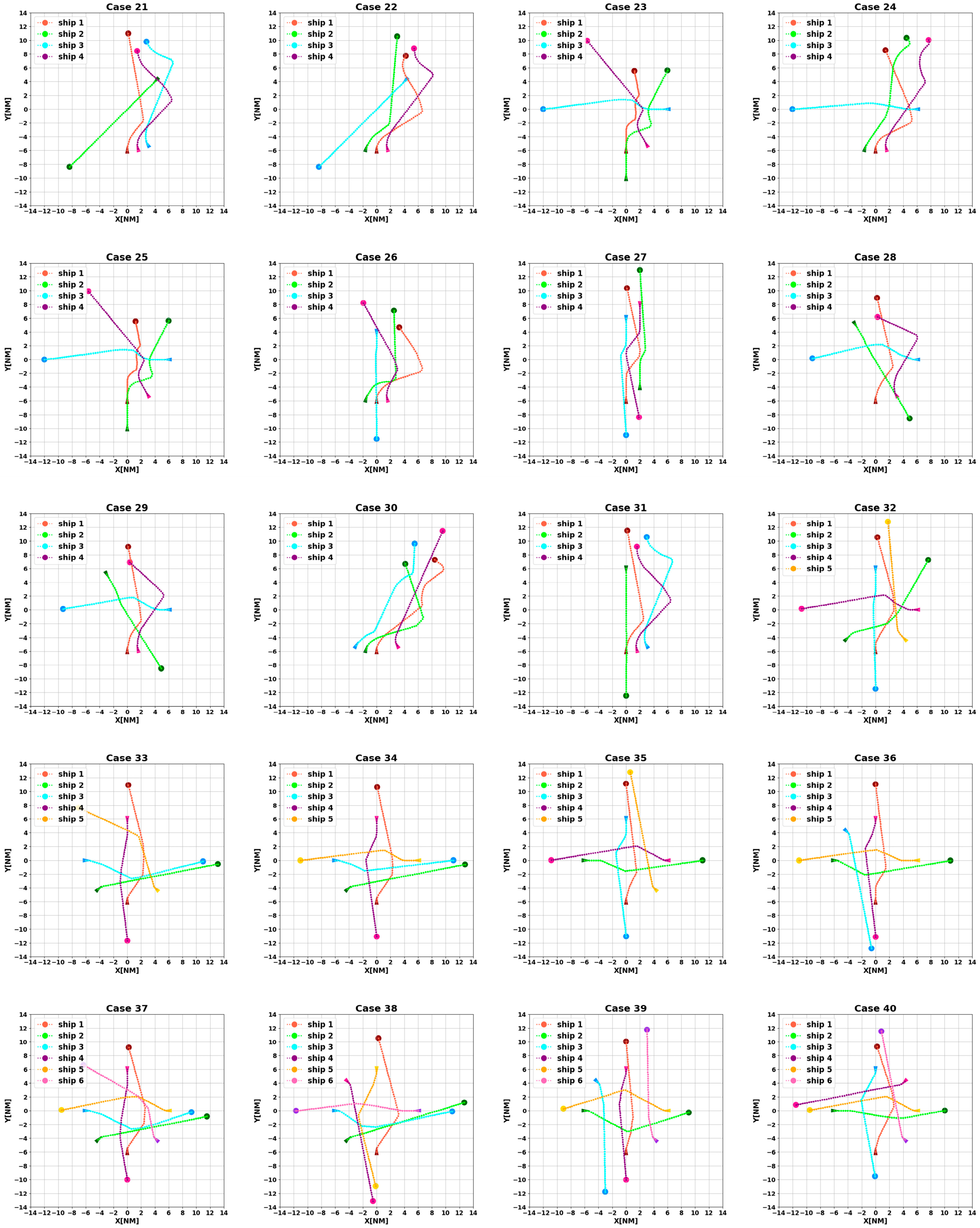
Appendix A

As stated in Section 4.2, the model test results are shown in Figures A1–A3. Panel (a) and Panel (b) of each figure, respectively, represent the test results of Case 1–20 and Case 21–40 in the extended encounter scenario library.



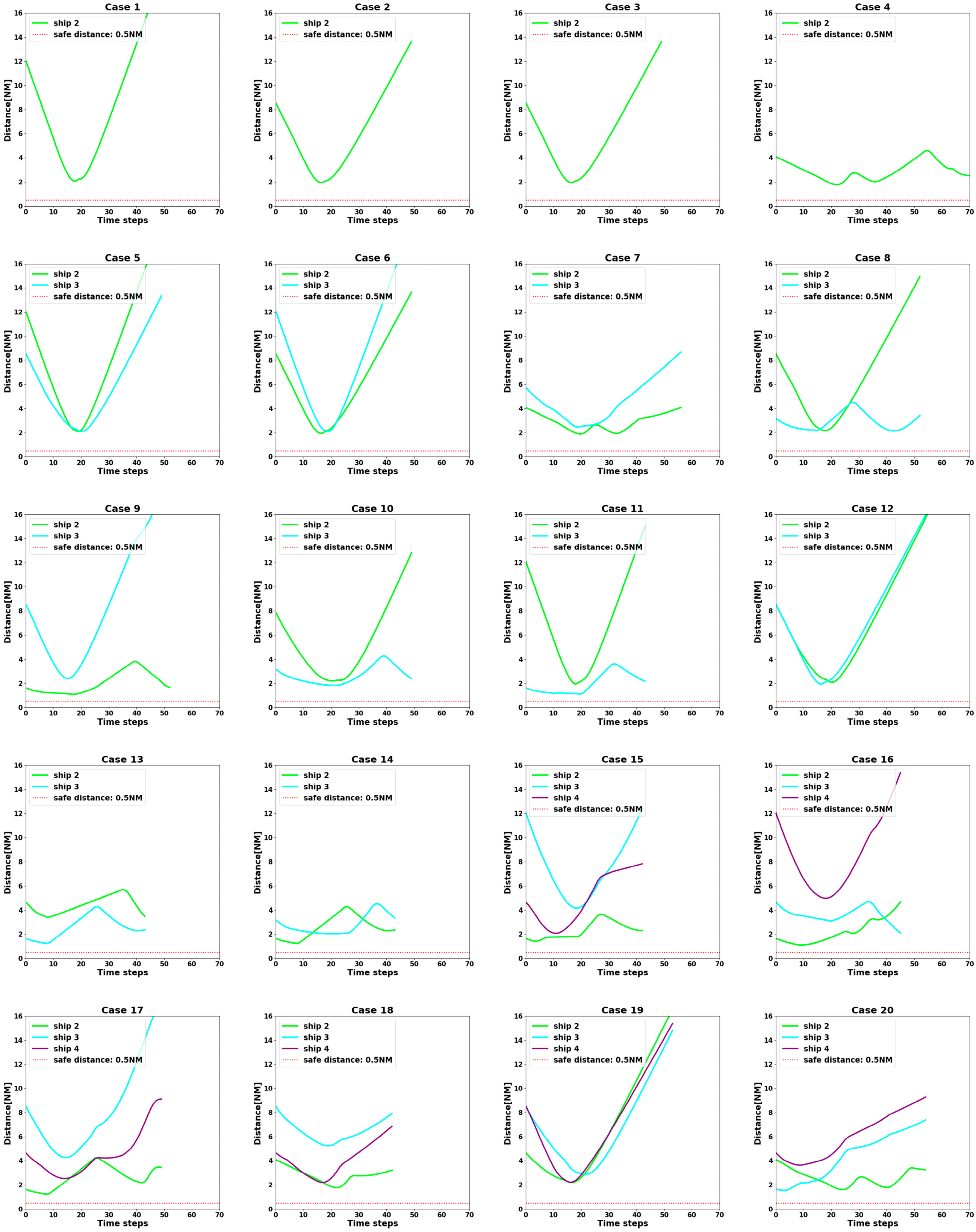
(**a**)

Figure A1. *Cont.*



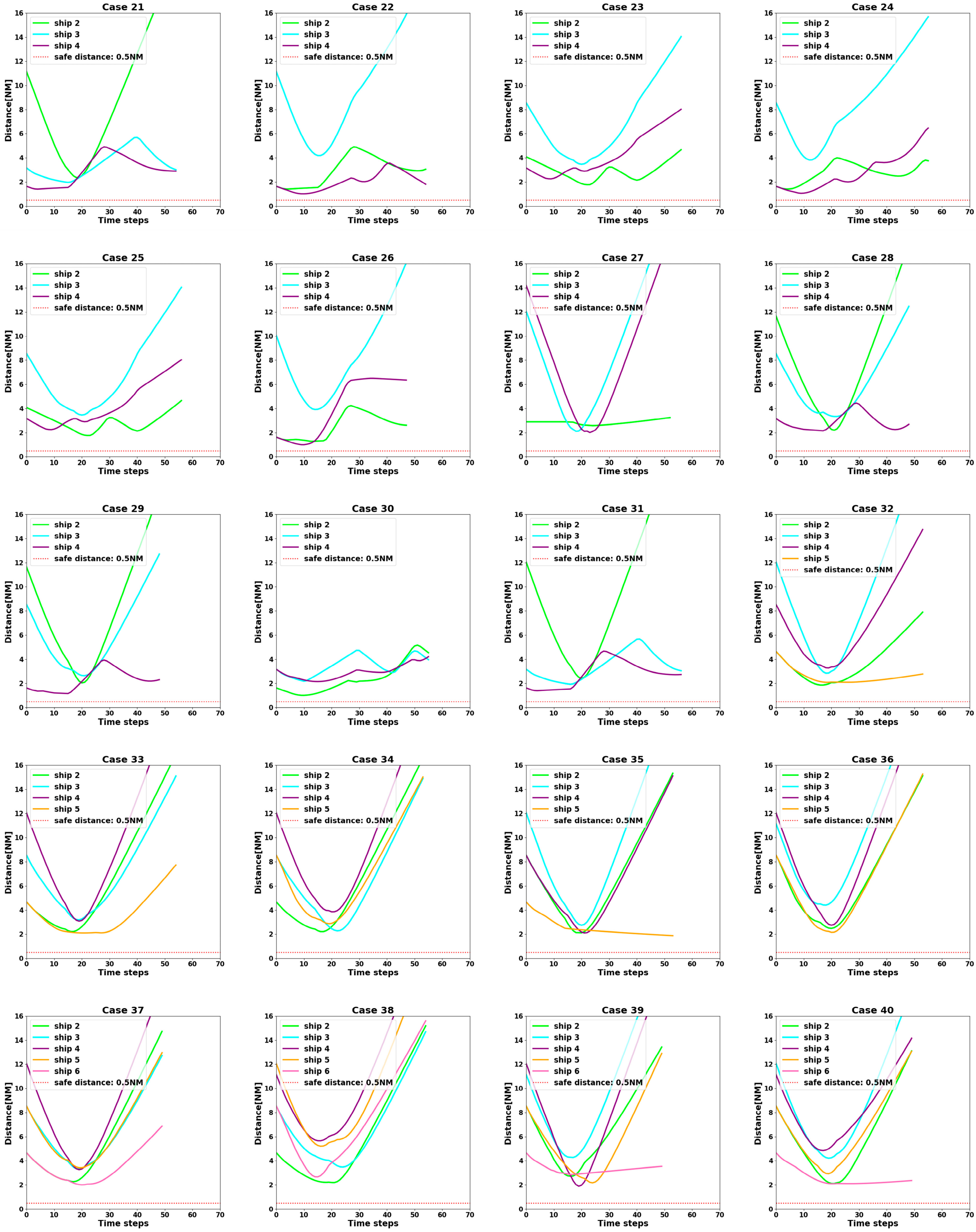
(**b**)

Figure A1. Ships' trajectories in the encounter scenario library. (**a**) Case 1–20; (**b**) Case 21–40.



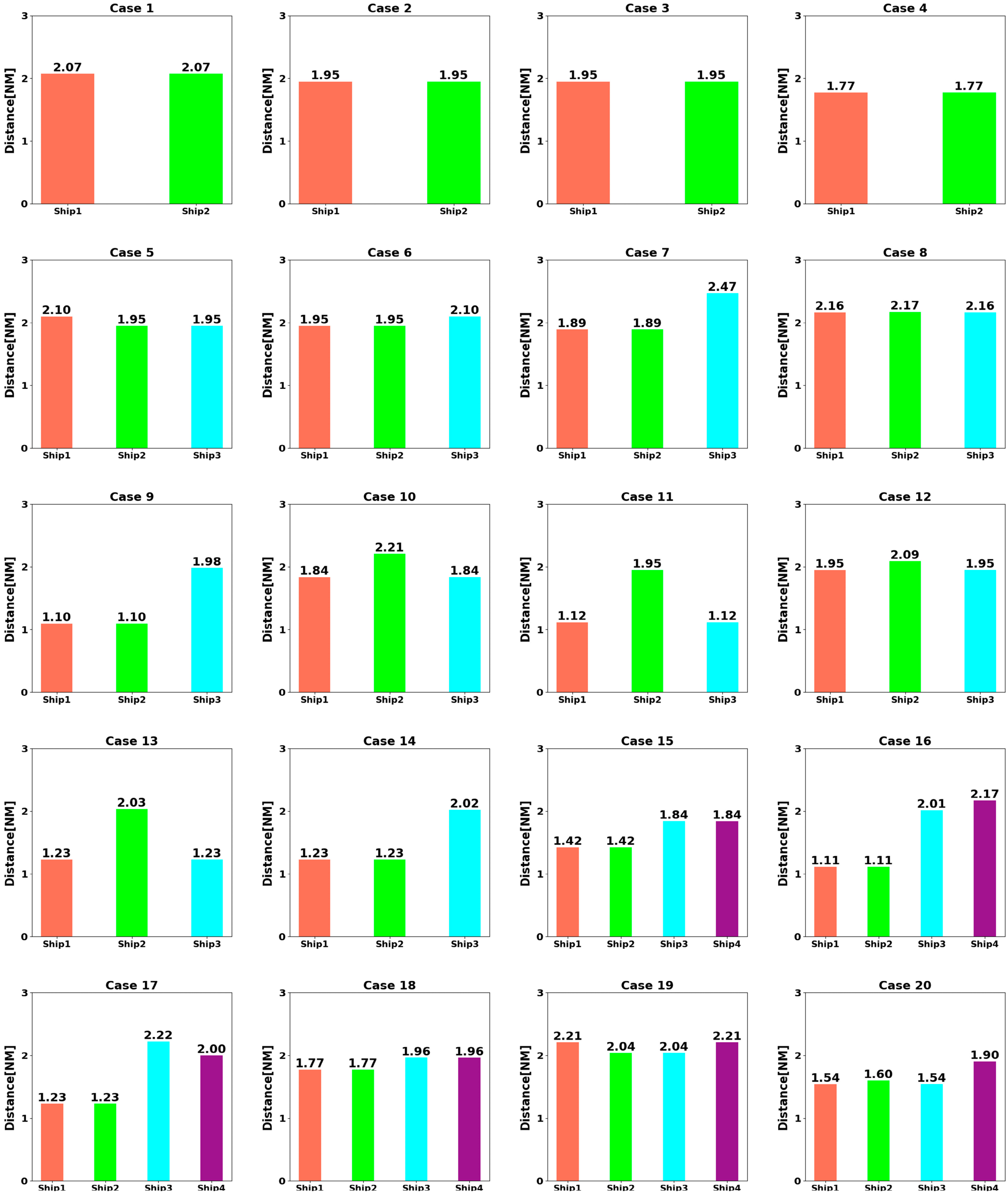
(**a**)

Figure A2. *Cont*.



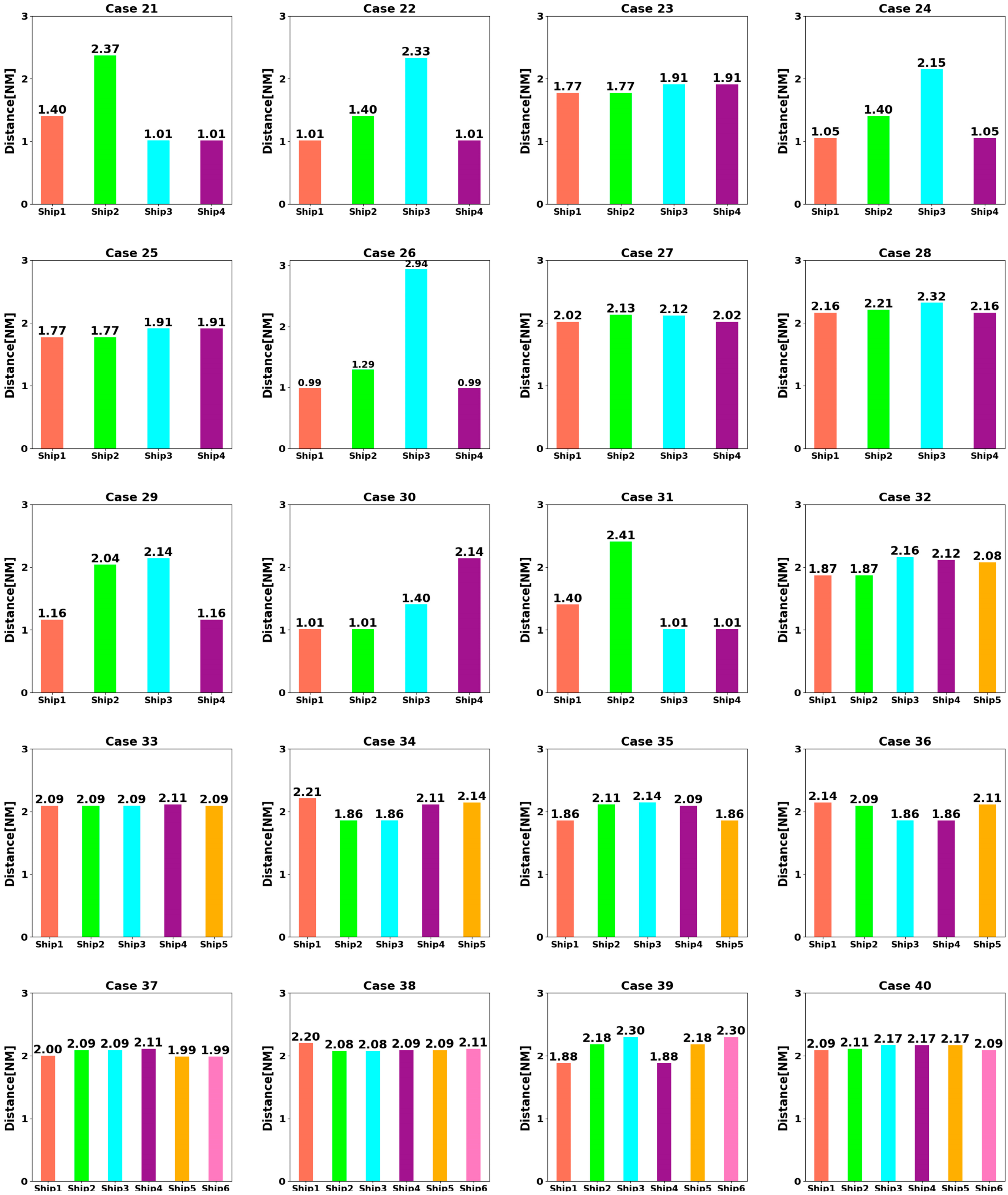
(**b**)

Figure A2. The changes in distance between the own ship (ship 1) and the target ships under this perspective. (**a**) Case 1–20; (**b**) Case 21–40.



(**a**)

Figure A3. *Cont.*



(**b**)

Figure A3. The minimum passing distance of each agent ship from other agent ships. (**a**) Case 1–20; (**b**) Case 21–40.

References

1. European Maritime Safety Agency. *Annual Overview of Marine Casualties and Incidents 2021*; EMSA: Lisbon, Portugal, 2021; Available online: https://www.emsa.europa.eu/newsroom/latest-news/item/4266-annual-overview-of-marine-casualties-and-incidents-2020.html (accessed on 11 August 2023).
2. Maritime Safety Committee. *Report of the Maritime Safety Committee on Its Ninety-Ninth Session*; IMO: London, UK, 2018; Available online: https://www.imo.org/en/MediaCentre/MeetingSummaries/Pages/MSC-99th-session.aspx (accessed on 16 August 2023).
3. Wei, G.; Kuo, W. COLREGs-Compliant Multi-Ship Collision Avoidance Based on Multi-Agent Reinforcement Learning Technique. *J. Mar. Sci. Eng.* **2022**, *10*, 1431. [CrossRef]
4. Zhang, Y.; Zhai, P. Research progress and trend of autonomous collision avoidance technology for marine ships. *J. Dalian Marit. Univ.* **2022**, *48*, 1–11.
5. Papadimitrakis, M.; Stogiannos, M.; Sarimveis, H.; Alexandridis, A. Multi-Ship Control and Collision Avoidance Using MPC and RBF-Based Trajectory Predictions. *Sensors* **2021**, *21*, 6959. [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. Ma, J.; Su, Y.; Xiong, Y.; Zhang, Y.; Yang, X. Decision-making method for collision avoidance of ships in confined waters based on velocity obstacle and artificial potential field. *China Saf. Sci. J.* **2020**, *30*, 60–66. [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. Ahn, J.-H.; Rhee, K.-P.; You, Y.-J. A study on the collision avoidance of a ship using neural networks and fuzzy logic. *Appl. Ocean Res.* **2012**, *37*, 162–173. [CrossRef]
11. Szłapczyński, R.; Ghaemi, H. Framework of an evolutionary multi-objective optimisation method for planning a safe trajectory for a marine autonomous surface ship. *Pol. Marit. Res.* **2019**, *26*, 69–79. [CrossRef]
12. Statheros, T.; Howells, G.; Maier, K.M.D. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *J. Navig.* **2008**, *61*, 129–142. [CrossRef]
13. Wang, C.; Zhang, X.; Cong, L.; Li, J.; Zhang, J. Research on Intelligent Collision Avoidance Decision-Making of Unmanned Ship in Unknown Environments. *Evol. Syst.* **2019**, *10*, 649–658. [CrossRef]
14. Sun, Z.; Fan, Y.; Wang, G. An Intelligent Algorithm for USVs Collision Avoidance Based on Deep Reinforcement Learning Approach with Navigation Characteristics. *J. Mar. Sci. Eng.* **2023**, *11*, 812. [CrossRef]
15. 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]
16. Sawada, R.; Sato, K.; Majima, T. Automatic Ship Collision Avoidance Using Deep Reinforcement Learning with LSTM in Continuous Action Spaces. *J. Mar. Sci. Technol.* **2021**, *26*, 509–524. [CrossRef]
17. Zhao, L.; Roh, M.-I. COLREGs-compliant multiship collision avoidance based on deep reinforcement learning. *Ocean Eng.* **2019**, *191*, 106436. [CrossRef]
18. Sutton, R.S.; McAllester, D.; Singh, S.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Adv. Neural Inf. Process. Syst.* **1999**, *12*, 1057–1063.
19. Luis, S.Y.; Reina, D.G.; Marin, S.L.T. A Multiagent Deep Reinforcement Learning Approach for Path Planning in Autonomous Surface Vehicles: The Ypacaraí Lake Patrolling Case. *IEEE Access* **2021**, *9*, 17084–17099. [CrossRef]
20. Chen, C.; Ma, F.; Xu, X.; Chen, Y.; Wang, J. A Novel Ship Collision Avoidance Awareness Approach for Cooperating Ships Using Multi-Agent Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2021**, *9*, 1056. [CrossRef]
21. Zhu, F.; Ma, Z. Ship trajectory online compression algorithm considering handling patterns. *IEEE Access* **2021**, *9*, 70182–70191. [CrossRef]
22. The International Maritime Organization (IMO). Convention on the International Regulations for Preventing Collisions at Sea (COLREGs). 1972. Available online: https://www.imo.org/fr/about/Conventions/Pages/COLREG.aspx (accessed on 21 August 2023).
23. Belcher, P. A sociological interpretation of the COLREGS. *J. Navig.* **2002**, *55*, 213–224. [CrossRef]
24. Zhu, F.; Zhou, Z.; Lu, H. Randomly Testing an Autonomous Collision Avoidance System with Real-World Ship Encounter Scenario from AIS Data. *J. Mar. Sci. Eng.* **2022**, *10*, 1588. [CrossRef]
25. Wang, X.; Zhang, Y.; Liu, Z.; Wang, S.; Zou, Y. Design of Multi-Modal Ship Mobile Ad Hoc Network under the Guidance of an Autonomous Ship. *J. Mar. Sci. Eng.* **2023**, *11*, 962. [CrossRef]
26. 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] [PubMed]
27. 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. [CrossRef]

28. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay. In Proceedings of the 4th International Conference on Learning Representations, San Juan, PR, USA, 2–4 May 2016. [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. Zhang, W.; Feng, X.; Qi, Y.; Shu, F.; Zhang, Y.; Wang, Y. Towards a model of regional vessel near-miss collision risk assessment for open waters based on AIS data. *J. Navig.* **2019**, *72*, 1449–1468. [CrossRef]
31. Yoo, Y.; Lee, J.-S. Evaluation of ship collision risk assessments using environmental stress and collision risk models. *Ocean Eng.* **2019**, *191*, 106527. [CrossRef]
32. 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. [CrossRef]
33. Fossen, T.I. *Guidance and Control of Ocean Vehicles*; John Wiley & Sons Inc.: Hoboken, NJ, USA, 1994.
34. Liu, J.; Zhao, B.; Li, L. Collision Avoidance for Underactuated Ocean-Going Vessels Considering COLREGs Constraints. *IEEE Access* **2021**, *9*, 145943–145954. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.